

Ατομική Διπλωματική Εργασία

**ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΤΟΥ ΒΙΟΛΟΓΙΚΟΥ
ΠΛΗΘΥΣΜΟΥ ELEONORA FALCON ΜΕ ΤΗ ΧΡΗΣΗ ΤΟΥ
ΜΟΝΤΕΛΟΕΛΕΚΤΗ UPPAAL SMC**

Κυριακός Ζαντής

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2017

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μοντελοποίηση και ανάλυση του βιολογικού πληθυσμού Eleonora Falcon με τη χρήση του μοντελοελεκτή Uppaal SMC

Κυριακός Ζαντής

Επιβλέπουσα Καθηγήτριας

Άννα Φιλίππου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των
απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του
Πανεπιστημίου Κύπρου

Μάιος 2017

Ευχαριστίες

Αρχίζοντας θα ήθελα να ευχαριστήσω την Δρ. Άννα Φίλιππου για την ασταμάτητη βοήθεια και υποστήριξη που μου παρείχε καθόλη την διάρκεια τις εκπνόνησης τις διπλωματικής εργασίας αλλά και κατά την διάρκεια τον μαθημάτων που μου δίδαξε. Πάντα μπορούσε να με καθοδηγήσει σωστά στη λύση του προβλήματος που αντιμετώπιζα δίνοντας μου σωστές κατευθυντήριες γραμμές, βοηθητικές συμβουλές και εύστοχους προβληματισμούς για να οδηγηθώ στην λύση. Ήταν και θα είναι πάντα μια από τους καλύτερους καθηγητές που μου έκαναν μάθημα στο Πανεπιστήμιο Κύπρου αφού με τον τρόπο που διδάσκει την ένιωθα κοντά μου, έχοντας την καλή θέληση να με βοηθήσει.

Επιπλέον θα ήθελα να ευχαριστήσω τους φίλους μου που με ενθάρρυναν να συνεχίσω με πάθος αυτή την εργασία ακόμα και όταν έφτανα σε αδιέξοδα. Ο ενθουσιασμός για την πρόοδο μου στην συγκεκριμένη εργασία, μου έδινε κουράγιο και με ενέπνεε να συνεχίσω. Επίσης ήταν έτοιμοι να με βοηθήσουν σε θέματα βιολογίας, μαθηματικών όσο και πληροφορικής σε περίπτωση που είχα κάποιο κενό για την σωστή διεκπεραίωση της διπλωματικής μου.

Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου που πάντα με στήριζαν σε κάθε επιλογή έκανα στην ζωή μου, οι συμβουλές τους, η αγάπη και συμπαράσταση που μου έδειξαν όλα αυτά τα χρόνια καθώς και το ήθος που με μεγάλωσαν έγιναν αυτό που είμαι σήμερα και γι αυτό τους ευχαριστώ.

Περίληψη

Η Μοντελοποίηση είναι η απεικόνιση πραγματικών φαινομένων σε κάποια γλώσσά που την διέπουν κάποιοι κανόνες και έτσι καθίστα το φαινόμενο να μπορεί να κατανοηθεί από κάποιο που γνωρίζει την γλώσσά αυτή [14]. Επίσης μπορούμε να χρησιμοποιήσουμε τεχνικές ή αλγορίθμους που υπάρχουν στην γλώσσά αυτή για να λύσουμε ή να μελετήσουμε τα φαινόμενα αυτά. Για παράδειγμα μοντελοποιώντας το την σύγκρουση ενός αυτοκίνητου μπορούμε να καταλάβουμε την αντίδραση που θα έχει το σώμα χωρίς ή με την ζώνη. Ακόμα και πιο περίπλοκα προβλήματα όπως το φαινόμενο του θερμοκηπίου μπορούν να μοντελοποιηθούν. Όλοι οι κλάδοι χρησιμοποιούν μοντέλα για πιο σωστή κατανόηση και μελέτη ενός αντικείμενου. Παραδείγματα είναι στη Φυσική η μοντελοποίηση του ηλιακού συστήματος, στη Μηχανολογία το μοντέλο μιας μηχανής αεροπλάνου, στην Αρχιτεκτονική το μοντέλο του ενός κτηρίου και στη Βιολογία το μοντέλο εξέλιξης ενός οργανισμού.

Τα μοντέλα στην Οικολογία είναι συνήθως προσομοιώσεις της συμπεριφοράς ενός πληθυσμού, με σκοπό να μελετήσουμε την εξέλιξη αυτού του πληθυσμού με την πάροδο του χρόνου ως προς το πλήθος του, αντιδράσεις σε μεταβολές στο περιβάλλον του ακόμα και μεταδόσεις ασθενειών. Υπάρχουν δυο βασικοί λόγοι που δημιουργούν την ανάγκη μοντελοποίησης ενός συγκεκριμένου πληθυσμού. Ο πρώτος λόγος είναι ότι πρέπει να αποτρέψουμε την εξαφάνιση αυτού του πληθυσμού είτε επειδή είναι ενδημικός είτε είναι προστατευόμενο είδος. Ο δεύτερος λόγος είναι ότι ο πληθυσμός αυτός αυξάνετε ραγδαία με αποτέλεσμα να απειλεί είτε την πανίδα είτε την χλωρίδα του γενικού οικοσυστήματος.

Αυτή η εργασία έχει στόχο τη μοντελοποίηση του Μαυροπετρίτη (Eleonora Falcon) το οποίο μεγάλο κομμάτι του παγκοσμίου πληθυσμού μαζεύεται και αναπαράγεται στα μεσογειακά νησιά, και κυρίως στην Ελλάδα, μεταναστεύοντας κάθε χρόνο στην Μαδαγασκάρη. Μετά από προηγούμενη μελέτη που έγινε στον Ελληνικό πληθυσμό και μοντελοποίηση στην PALPS και PRISM [14] αποφάσισα να κάνω το αντίστοιχο με τον Κυπριακό πληθυσμό, μελετώντας βέβαια και αλλά εργαλεία όπως το UPPAAL SMC, για την βελτιστοποίηση της διαδικασίας αυτής και μελέτη μεγαλύτερων πληθυσμών για πιο μεγάλο χρονικό διάστημα.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή	1
1.1	Κίνητρο	1
1.2	Στόχοι εργασίας	2
1.3	Μεθοδολογία και αποτελέσματα	3
1.4	Οργάνωση κεφαλαίων	3
Κεφάλαιο 2	Προηγούμενη Εργασία	5
2.1	Μοντελοποίηση και Προσομοίωση	5
2.1.1	Προσομοίωση Συμπεριφοράς Αφαιρετικών Μοντέλων	7
2.1.2	Εφαρμογές Δυναμικών Αφαιρετικών Μοντέλων	8
2.2	Μοντελοποίηση Οικολογικών Συστημάτων	9
2.3	Έλεγχος Συστημάτων	11
2.4	Θεωρία Αυτομάτων	12
2.5	Μοντελοέλεγχος	14
2.5.1	Χρονικές και Τροπικές Λογικές	15
2.5.2	Προτασιακή Γραμμική Χρονική Λογική	16
2.5.3	Προτασιακή Διακλαδωμένη Χρονική Λογική	16
2.5.4	Στατιστικός Μοντελοέλεγχος	17
Κεφάλαιο 3	Μελέτη εργαλείων Μοντελοέλεγχου	19
3.1	Περιγραφή Parasitic Varroa mite	19
3.2	Μελέτη εργαλείου PRISM	20
3.2.1	Περιγραφή Εργαλείου PRISM	20
3.2.2	Μοντελοποίηση Parasitic Varroa mite	21
3.2.3	Αποτελέσματα	23
3.2.4	Συμπεράσματα	23
3.3	Μελέτη εργαλείου UPPAAL	24
3.3.1	Περιγραφή Εργαλείου UPPAAL SMC	24
3.3.2	Μοντελοποίηση Parasitic Varroa mite	25
3.3.3	Μοντελοποίηση Dynamic Template στο UPPAAL SMC	27

3.3.4	Αποτελέσματα	28
3.3.5	Συμπεράσματα	30
3.4	Επιλογή εργαλείου UPPAAL SMC	30
Κεφάλαιο 4	Ο Μαυροπετρίτης και Ανάλυση Δεδομένων	32
4.1	Μαυροπετρίτης	32
4.2	Ανάλυση των δεδομένων	34
Κεφάλαιο 5	Δημιουργία και Ανάλυση Μοντέλου	37
5.1	Δημιουργία μοντέλου	37
5.2	Δημιουργία Μοντέλου με Μέγιστο Αριθμό πληθυσμού	37
5.3	Δημιουργία Μοντέλου με Δυναμική Δημιουργία στοιχείων	42
5.4	Ρύθμιση πιθανοτήτων για ανταπόκριση στην πραγματικότητα	45
5.5	Αποτελέσματα	47
Κεφάλαιο 6	Συμπεράσματα	51
6.1	Αξιολόγηση Αποτελεσμάτων	51
6.2	Μελλοντικές Εργασίες	52
Βιβλιογραφία		53
Παράρτημα Α		A-3
Παράρτημα Β		B-7
Παράρτημα C		C-6
Παράρτημα D		D-16
Παράρτημα Ε		E-11
Παράρτημα F		F-1

Κεφάλαιο 1

Εισαγωγή

1.1 Κίνητρο	1
1.2 Στόχοι εργασίας	2
1.3 Μεθοδολογία και αποτελέσματα	3
1.4 Οργάνωση κεφαλαίων	3

1.1 Κίνητρο

Ζώντας σε μια εποχή όπου οι κλιματολογικές αλλαγές συμβαίνουν ραγδαία αυξάνοντας την μέση θερμοκρασία, πολλοί φυσικοί βιότοποι έχουν άμεση επιρροή από αυτό όπως το λιώσιμο των πάγων στους πόλους τις γης. Υπάρχουν βεβαία και βιότοποι που η αλλαγή έρχεται στην πάροδο του χρόνου. Αλλάζοντας τους βιοτόπους επηρεάζονται χιλιάδες οργανισμοί που ζουν τρέφονται και αναπαράγονται. Αφού αυτοί οι οργανισμοί είναι μέρος τις τροφικής αλυσίδας του πλανήτη, όπως και εμείς, η ζωτικότητα τους μας επηρεάζει και εμάς έμμεσα. Βέβαια οι πληθυσμοί αυτοί έχουν πολλούς παράγοντες που επηρεάζουν την ζωτικότητα τους όπως άλλοι πληθυσμοί που μπορεί να αποτελούν τους κυνηγούς ή τα θηράματα του πληθυσμού αυτού ή ακόμα και συναγωνιστές στην τροφή και τόπου κατοίκησης. Επίσης και ο ανθρώπινος παράγοντας τους επηρεάζει καταστρέφοντας τον βιότοπο που ζουν με πρόφαση την τεχνολογική ανάπτυξη ή και άμεσα σκοτώνοντας τον πληθυσμό σε οποιαδήποτε φάση της ζωής του.

Η Οικολογία είναι ο κλάδος της Βιολογίας και μελετά τους ζωντανούς οργανισμούς, την αλληλεπίδραση που έχουν μεταξύ τους και με άλλους οργανισμούς και το περιβάλλον γύρω τους, θέλει να είναι σε θέση να απαντήσει άμεσα στα ερωτήματα που αφορούν τα μετρά πρέπει να παρθούν για την αποφυγή εξελίξεις ειδών και καταστροφή του οικοσυστήματος μας. Για να παρέχουμε προστασία σε τέτοιους οργανισμούς, πρέπει πρώτα να τους μελετήσουμε και να τους κατανοήσουμε. Η

συλλογή δεδομένων και η ανάλυση τους από την Επιστήμη της Οικολογίας είναι χρονοβόρα αφού ο κύκλος ζωής των οργανισμών αυτών μερικές φορές μπορεί να είναι και χρόνια.

Εδώ έρχεται να τοποθετήσει το λιθαράκι της η Επιστήμη της Πληροφορικής στην προσπάθεια αυτή αφού με την κατασκευή του σωστού μοντέλου του κύκλου ζωής ενός ή περισσότερων οργανισμών μπορεί να τους προσομοιώσει στο βάθος του χρόνου προσφέροντας με μεγάλη πιθανότητα σωστές εκτιμήσεις. Αλλάζοντας κάποιους παράγοντες που επηρεάζουν την ζωή του οργανισμού στο μοντέλο, όπως για παράδειγμα η θνησιμότητα από τον ήλιο, μπορούμε να υπολογίσουμε σε διάστημα κάποιων χρόνων πως θα εξελιχτεί ο πληθυσμός του οργανισμού αυτού.

Ο Μαυροπετρίτης αποτελεί ένα ενδιαφέρον είδος προς μοντελοποίηση αφού ενώ τον χειμώνα ζει στον Ινδικό Ωκεανό και στην Ανατολική Αφρική, ένα μεγάλο μέρος του παγκόσμιου πληθυσμού μαζεύεται τους καλοκαιρινούς μήνες στα λίγα μεσογειακά νησιά, εκ των οποίων το 80% περίπου στα νησιά του Αιγαίου Πελάγους, για να κάνει αναπαραγωγή. Παρόλο που παγκόσμια ο πληθυσμός αυτός δεν απειλείται, στην Ελλάδα θεωρείται σαν σπάνιο αφού η διαβίωση του εξαρτάται από το αν στο νησί υπάρχουν οι κατάλληλες συνθήκες για να γίνει η αναπαραγωγή του. Κατ' ακρίβεια η περίοδος αναπαραγωγής του συμπίπτει με τους καλοκαιρινούς μήνες όπου οι κλιματικές αλλαγές έχουν άμεση επίπτωση στην αναπαραγωγή τους καθώς παράλληλα είναι η ακμή της τουριστικής περιόδου στα νησιά της Μεσόγειου.

1.2 Στόχοι εργασίας

Βασισμένος σε προηγούμενη ερευνά που είχε σκοπό μοντελοποίηση του ελληνικού πληθυσμού του Μαυροπετρίτη σε σύνταξη της γλώσσας PALPS και μεταφορά των εξισώσεων στον μοντελοελεκτή Prism [14], στόχος της διπλωματικής αυτής είναι η μοντελοποίηση του αντιστοίχου Κυπριακού Πληθυσμού. Η ερευνά αυτή έδειξε πως ο άνθρωπος μπορεί να συμβάλει στην προστασία και διατήρηση του Ελληνικού πληθυσμού του Μαυροπετρίτη, με την θήρευση ποντίκων ή από την μείωση καταστροφής από τον ανθρώπινο παράγοντα. Στόχος επίσης της διπλωματικής αυτής ήταν η βελτίωση της μεθοδολογίας που χρησιμοποιήθηκε, μελετώντας περισσότερα εργαλεία Μοντελοέλεγχου, έτσι να μπορέσει ο πληθυσμός να μοντελοποιηθεί για

μεγαλύτερο αριθμό πουλιών και για περισσότερα χρόνια. Μοντελοποιώντας την συμπεριφορά του Μαυροπετρίτη, με βάση των στατιστικών δεδομένων που πάρθηκαν από το Τμήμα Βιολογίας, έγινε στην συνέχεια η προσαρμογή στην πραγματικότητα. Χρησιμοποιώντας δηλαδή τα δεδομένα από τα πρώτα χρόνια προσπαθούμε να προσομοιώσουμε την συμπεριφορά για τα τελευταία χρόνια που έχουμε τα δεδομένα, συγκρίνοντας αν υπάρχει αποδεκτή απόκλιση και κάνοντας τις κατάλληλες ρυθμίσεις στο μοντέλο σε περίπτωση που πάρουμε λάθος δεδομένα. Αυτό οδήγησε στο να αντιληφθούμε την συμπεριφορά του καλύτερα ανακαλύπτοντας τις παραμέτρους που συμβάλουν περισσότερο στη ζωή του και κατανόηση των κινδύνων που των απειλούν. Τέλος απομονώνοντας τον κάθε κίνδυνο ξεχωριστά μελετήθηκε πόσο επηρεάζει ο κίνδυνος αυτός και αν είναι πιθανό να μπορεί να πάρει μέτρα ο άνθρωπος έτσι ώστε να το προστατεύσει

1.3 Μεθοδολογία και αποτελέσματα

Πρώτο βήμα για την επίτευξη των στόχων μου ήταν η μελέτη διάφορων μοντελοελεγκτών και να εντοπιστεί αυτός που παρείχε τις περισσότερες δυνατότητες για την δημιουργία μοντέλου του πληθυσμού. Μελετήθηκε ο μοντελοελεγκτής PRISM με την προσομοίωση ενός μικρού πληθυσμού και ο μοντελοελεγκτής UPPAAL, και συγκεκριμένα ο μοντελοελεγκτής UPPAAL SMC ο οποίος παρέχει στατιστικό έλεγχο μοντέλου, όπου και επιλέχτηκε ο δεύτερος. Στην συνέχεια έγινε η δημιουργία μοντέλου του Μαυροπετρίτη στο UPPAAL SMC με βάση τα δεδομένα που είχαμε στην κατοχή μας από τον αντίστοιχο Ελληνικό πληθυσμό. Κατόπιν σε συνεργασία με το τμήμα Βιολογικών Επιστημών του Πανεπιστημίου Κύπρου, συγκεκριμένα με τον Δρ. Alexander N. G. Kirschel και τον διδακτορικό φοιτητή Θωμά Χατζηκυριάκου, έγινε η ρύθμιση των δεδομένων με τα κυπριακά δεδομένα και η αξιολόγηση του μοντέλου κατά πόσο είναι ορθό.

1.4 Οργάνωση κεφαλαίων

Στο Κεφάλαιο 2 παρουσιάζονται οι σχετικές εργασίες με την διπλωματική μου που αφορούν την μοντελοποίηση, τα είδη της μοντελοποίησης, εργασίες που αφορούν την μοντελοποίηση των οικολογικών πληθυσμών επεξήγηση των αυτομάτων και τι είναι ο

μοντελοέλεγχος. Στο Κεφάλαιο 3 θα μελετηθούν οι 2 μοντελοελεκτές PRISM και UPPAAL SMC με σκοπό να βρεθεί ο μοντελοελεκτήρας που καλύπτει τις ανάγκες της μοντελοποίησης του πληθυσμού Κυπριακού πληθυσμού του Μαυροπετρίτη. Στο Κεφάλαιο 4 θα γίνει επεξήγηση της ζωής του Μαυροπετρίτη και η ανάλυση των δεδομένων που πήραμε από το Τμήμα Βιολογίας, εκφράζοντας έτσι τις πιθανότητες που αναπαράγεται και πεθαίνει στον κύκλο ζωής του ο Μαυροπετρίτη. Στο Κεφάλαιο 5 με βάση την ανάλυση που έγινε στο προηγούμενο κεφάλαιο θα παρουσιαστεί η δημιουργία του μοντέλου σε κάθε φάση στο κύκλο ζωής του πληθυσμού και πρόταση 2 προσεγγίσεων, η μια είναι η μοντελοποίηση με προκαθορισμένο μέγιστο αριθμό και η άλλη είναι μοντελοποίηση με δυναμική δημιουργία στοιχείων του πληθυσμού αυτού. Θα παρουσιαστούν τα δεδομένα που βγάζουν οι πιθανότητες όπως τις πήραμε και η ρύθμιση τους με βάση τα πραγματικά δεδομένα. Τέλος στο Κεφάλαιο 6 θα παρουσιαστούν τα δεδομένα που εξήγαγαν οι προσομοιώσεις του μοντέλου και θα αναλυθούν τα συμπεράσματα της διπλωματικής μαζί με τους στοχασμούς για μελλοντικές εργασίες

Κεφάλαιο 2

Προηγούμενη Εργασία

2.1 Μοντελοποίηση και Προσομοίωση	5
2.1.1 Προσομοίωση Συμπεριφοράς Αφαιρετικών Μοντέλων	7
2.1.2 Εφαρμογές Δυναμικών Αφαιρετικών Μοντέλων	8
2.2 Μοντελοποίηση Οικολογικών Συστημάτων	9
2.3 Έλεγχος Συστημάτων	11
2.4 Θεωρία Αυτομάτων	12
2.5 Μοντελοέλεγχος	14
2.5.1 Χρονικές και Τροπικές Λογικές	15
2.5.2 Προτασιακή Γραμμική Χρονική Λογική	16
2.5.3 Προτασιακή Διακλαδωμένη Χρονική Λογική	16
2.5.4 Στατιστικός Μοντελοέλεγχος	17

2.1 Μοντελοποίηση και Προσομοίωση

Ο κόσμος γύρω μας αλλάζει καθημερινά με ένα απρόβλεπτο ρυθμό. Νέα γεγονότα παρουσιάζονται φέρνοντας αλλαγές στην καθημερινότητα μας τόσο σε μεγάλο βαθμό όσο και σε μικρό. Τα γεγονότα αυτά είναι μερικές φορές τυχαία γεγονότα από την φύση ή γεγονότα τα οποία συμβήκανε λόγω της επιρροής από τον άνθρωπο. Αυτές τις δυναμικές αλλαγές το μυαλό του ανθρώπου δεν είναι σε θέση να τις καταλάβει και να τις υπολογίσει πλήρως. Συνήθως τις αντιλαμβανόμαστε όταν είναι πολύ αργά. Οι δυναμικές διεργασίες, όπως ονομάζει τα γεγονότα αυτά ο συγγραφέας στο [11], αποτελούνται από ένα ευρύ φάσμα παραδειγμάτων όπως συστήματα οδικής κυκλοφορίας, η οικονομική εξέλιξη μιας χώρας, η ανάπτυξη μιας αρρώστιας σε ένα οργανισμό, ο τρόπος που μεταφέρεται μια είδηση και πως μπορεί να αλλοιωθεί,

οικολογικοί βιότοποι και οργανισμοί που ζουν σε αυτούς αλλά και πιο επικίνδυνα όπως πόλεμοι και η διάδοση μιας θανατηφόρας αρρώστιας σε ένα πληθυσμό.

Για να αποτρέψουμε ή να προτρέψουμε κάποιες από αυτές τις δυναμικές διεργασίες, πρέπει να ανακαλύψουμε τις συνθήκες που τις διέπουν οι οποίες ακόμα και σε μικρά συστήματα μπορεί να είναι αρκετά δύσκολες [11]. Ένα μοντέλο θα πρέπει να προσομοιώνει την συμπεριφορά μιας δυναμικής διεργασίας, δίνοντας πιθανά σενάρια που μπορούν να συμβούν και υπό ποιες συνθήκες θα συμβεί το κάθε ένα, έτσι ώστε να μπορούμε να πάρουμε τα κατάλληλα μέτρα για το επιθυμητό αποτέλεσμα.

Μοντελοποίηση

Μοντελοποίηση είναι η απλή αναπαράσταση των πραγματικών γεγονότων αντιγράφοντας τα βασικά στοιχεία αυτού που θέλεις να μοντελοποιηθεί. Μια μοντελοποίηση μπορεί να είναι μαθηματική εξίσωση, μοντελοποίηση στον υπολογιστή, φυσική μοντελοποίηση ή κάποιος συνδυασμός τους [1].

Προσομοίωση

Ακούγοντας την λέξη προσομοίωση μας έρχεται στο μυαλό ότι έχουμε να κάνουμε με το ομοίωμα κάποιου αντικειμένου, δηλαδή μίμηση ή αντιγραφή. Για παράδειγμα θα μπορούσαμε να πούμε ότι πολλά παιδικά παιχνίδια είναι προσομοιώσεις πραγματικών καταστάσεων όπως το παιχνίδι «Κατάστημα» που προσομοιώνει τις σχέσεις πελάτη πωλητή απλά και κατανοητά για τα μάτια των παιδιών.

Σύστημα

Ένα σύστημα με βάση το [1] θεωρείται μια συλλογή από αντικείμενα, κομμάτια ή στοιχεία τα οποία αλληλεπιδρούν μεταξύ τους με κάποιους χωρικούς περιορισμούς δημιουργώντας ένα συγκεκριμένο μοτίβο συμπεριφοράς. Περιορίζοντας το χωρικά μπορείς να το απομονώσεις από τον υπόλοιπο κόσμο και να το μελετήσεις.

Φυσικά Μοντέλα

Είναι τα μοντέλα που αποτελούνται από φυσικά αντικείμενα, αντίγραφα των πραγματικών, με σκοπό να αναπαριστούν τα πραγματικά. Δημιουργώντας μοντέλα αυτοκινήτων οι μεγάλες αυτοκινητοβιομηχανίες, μπορούν να προσομοιώσουν την

αντίδραση που θα έχει το αυτοκίνητο σε ακραία φαινόμενα όπως σύγκρουση σε μεγάλες ταχύτητες και πόσο γρήγορα μπορούν να ανταποκριθούν τα φρένα.

Αφαιρετικά Μοντέλα

Είναι τα μοντέλα που αποτελούνται από τις μαθηματικές εξισώσεις και τις μοντελοποιήσεις σε υπολογιστικά συστήματα. Για να κατασκευάσει κάποιος ένα τέτοιο μοντέλο πρέπει να εκφράσει την συμπεριφορά του συστήματος όπως αυτός την αντιλαμβάνεται είτε από αρχικά και τελικά δεδομένα είτε κατατάσσοντας τις φάσεις της ζωής του συστήματος σε καταστάσεις. Ονομάζονται αφαιρετικά αφού κάποια δεδομένα μπορούν να εξηγηθούν με πιο γενικές έννοιες, όπως για παράδειγμα οι αιτίες θνησιμότητας κάποιου πληθυσμού να μεταφραστούν όλες με ένα γενικό ποσοστό, με σκοπό την καλύτερη κατανόηση ή την ανεπάρκεια πόρων.

2.1.1 Προσομοίωση Συμπεριφοράς Αφαιρετικών Μοντέλων

Για να μπορούμε να κάνουμε προσομοίωση μια συμπεριφοράς έχουμε 2 διαφορετικές μεθόδους [11].

Η πρώτη μέθοδος η οποία αποτελείται από παρατηρήσεις και δεδομένα που αφορούν ένα σύστημα ή πολλά πανομοιότυπα συστήματα είναι η **Περιγραφή Συμπεριφοράς**. Στην περιγραφή συμπεριφοράς έχουμε δεδομένα από αρχικές καταστάσεις και πως άλλαξαν αυτά στην πάροδο του χρόνου ή από την εφαρμογή κάποιων ενεργειών, συγκρίνοντας τα με δεδομένα από τελικές καταστάσεις. Από αυτά δημιουργούμε μαθηματικές σχέσεις οι οποίες μπορεί να μην έχουν κάποια σχέση με τις πραγματικές ενέργειες αλλά να έχουν το ίδιο αποτέλεσμα. Αντιμετωπίζουμε το σύστημα σαν ένα “μαύρο κουτί” αφού μας ενδιαφέρει μόνο η είσοδος και η έξοδος του [11].

Η δεύτερη μέθοδος είναι η **Εξήγηση Συμπεριφοράς**, η οποία περιγράφει τις πραγματικές καταστάσεις του συστήματος και σε τι κατάσταση μεταβάλετε σε περίπτωση που συμβεί κάποια κανονική ενέργεια που θα έκανε το σύστημα. Για να γίνει αυτό πρέπει να γνωρίζουμε την σύσταση του συστήματος και πως επηρεάζει το κάθε μέρος του συστήματος το άλλο. Το σύστημα αυτό ονομάζεται “γυάλινο κουτί” [1] αφού μπορείς να δεις τα στοιχειά που διέπουν την συμπεριφορά του.

2.1.2 Εφαρμογές δυναμικών αφαιρετικών μοντέλων

Τα πεδία που μπορούν να εφαρμοστούν οι δυναμικές προσομοιώσεις αφαιρετικών μοντέλων είναι αρκετές αλλά μπορούν να κατηγοριοποιηθούν σε 4 γενικές κατηγορίες [11].

- Επιστημονική κατανόηση
- Ανάπτυξη συστημάτων στην τεχνολογία
- Διαχείριση συστήματος
- Δημιουργία Σχεδιασμού

Η Επιστημονική κατανόηση έχει σκοπό να μελετήσει και να οδηγήσει το μελετητή της σε μια νέα άγνωστη από την αρχική γνώση για το σύστημα που δεν φαινόταν άμεσα από αυτή. Μπορεί να ανακαλύψει για ολικές καταστροφικές αλλαγές στο σύστημα με μια μικρή αλλαγή στις αρχικές παραμέτρους. Συνήθως δεν είναι καθαρά ορατές οι δευτερογενείς και τριτογενείς σχέσεις μεταξύ των στοιχείων του συστήματος. Ένα παράδειγμα όπως περιγράφεται στο [11] είναι η συμπεριφορά των δασών όταν υποστούν την ατμοσφαιρική μόλυνση. Τα δέντρα του συστήματος είναι αρκετά ευαίσθητα και κάνοντας μια μικρή αλλαγή στη ατμοσφαιρική μόλυνση οδηγεί στη καταστροφή του συστήματος, αν και δεν φαινόταν αν μελετούσες το κάθε δέντρο ατομικά και τις άμεσες σχέσεις του με το περιβάλλον.

Η Ανάπτυξη συστημάτων στην τεχνολογία είναι για μοντελοποίηση ηλεκτρονικών συστημάτων. Μιας και τα συστήματα αυτά μπορούν να μοντελοποιηθούν σχεδόν με απόλυτη ακρίβεια, μιας και μπορούν να περιγράψουν με μαθηματικά μοντέλα, έχουν σκοπό να βελτιώσουν τα συστήματα αυτά. Μπορούμε για παράδειγμα να κάνουμε την προσομοίωση ενός επεξεργαστή και να εφαρμόζουμε αλγορίθμους για βελτιστοποίησης της χρήσης της βοηθητικής μνήμης τρέχοντας σε αυτό Benchmarks, χωρίς να σπαταλούμε πόρους και χρόνο, με τα ίδια ακριβώς αποτελέσματα που θα παίρναμε στο κανονικό σύστημα. Επίσης μπορούν να συμπεριληφθεί σε αυτή την κατηγορία προσομοιωτές δικτύων με σκοπό την μελέτη στην εφαρμογή νέων αλγορίθμων στις επικοινωνίες των κόμβων του δικτύου.

Η Διαχείριση συστήματος περιλαμβάνει τη μοντελοποίηση ενός συστήματος με το πραγματικό σύστημα παράλληλα και καθώς αυτό εξελίσσεται ανατροφοδοτείται και αλλάζει. Τρέχοντας πραγματικά δεδομένα παράλληλα, το μοντέλο μπορεί να προβλέψει κρίσιμες καταστάσεις που πρέπει να παρθούν άμεσες ενέργειες και το αντίκτυπο τους. Αποτελείται συνήθως από μοντέλα προσομοίωσης στη γεωργία με σκοπό να προσομοιώσουν τις γρήγορες αλλαγές στο εύφορο έδαφος που συμβαίνουν με τις μεταβολές των υδάτινων μαζών.

Η Δημιουργία Σχεδιασμού αφορά κυρίως την μελέτη μακροχρόνιων κοινωνικών αλλαγών από την αλληλεπίδραση διαφορετικών κοινωνικών ομάδων. Είναι μοντέλα που περιλαμβάνουν υποθέσεις και θεωρίες παιγνίων. Παράδειγμα είναι η μοντελοποίηση μεταφοράς μεγάλης μάζας προσφύγων από αραβικές χώρες στην Ευρώπη, με σκοπό να μελετήσουμε το αντίκτυπο που θα έχει η αναγκαστική αυτή «συγκατοίκηση» και στις 2 αυτές διαφορετικές κοινωνικά και θρησκευτικά ομάδες.

2.2 Μοντελοποίηση Οικολογικών συστημάτων

Ένα είδος που τράβα αρκετά το ενδιαφέρον των επιστημόνων, μιας και το αντίκτυπο αυτών των συστημάτων έχει άμεση σημασία στην βιωσιμότητα κόσμου γύρω μας, είναι τα οικολογικά συστήματα. Οικολογικό σύστημα ονομάζουμε το περιβάλλον που περιλαμβάνει όλους τους οργανισμούς και όλες τις λειτουργικές διεργασίες που κάνουν το περιβάλλον αυτό, βιώσιμο για εκείνους [7]. Πρέπει να ξέρουμε το άμεσο μέλλον του φυσικού μας κόσμου για να μπορούμε να πάρουμε τα σωστά μέτρα και στρατηγικές έτσι ώστε να καταφέρουμε να αποτρέψουμε μια επικείμενη οικολογική καταστροφή. Ξέρουμε ήδη από μοντέλα που εφαρμοστήκαν για τις κλιματολογικές αλλαγές ότι η μέση θερμοκρασία τα επόμενα 100 χρόνια θα αυξηθεί μέχρι και 4 βαθμούς Κελσίου.

Οι μοντελοποιήσεις οικολογικών πληθυσμών είναι αναγκαστικά αφαιρετικά μοντέλα, αφού δεν μπορείς να μοντελοποιήσεις με φυσικά μοντέλα τους οργανισμούς αυτούς, καθώς αυτό θα έπαιρνε χρόνια και θα ήταν άσκοπη η μοντελοποίηση αφού θα είχαμε τα κανονικά αποτελέσματα. Επίσης ίσως μερικές φορές θα ήταν ανήθικη μια τέτοια μοντελοποίηση αφού μπορεί να θέλουμε να μελετήσουμε το αντίκτυπο μιας θανατηφόρας ασθένειας σε ένα πληθυσμό. Τα στοιχεία των μοντελοποιήσεων

οικολογικών πληθυσμών είναι συνήθως στο ατομικό επίπεδο των οργανισμών που μελετούμε, δηλαδή η κάθε κατάσταση που βρίσκεται το στοιχείο αντιστοιχεί σε μια κανονική φάση της ζωής του οργανισμού. Μερικές φορές οι οργανισμοί αυτοί μπορούν να μοντελοποιηθούν μαζικά, για παράδειγμα αν θέλουμε να δούμε την επιρροή των ακριδών στην γεωργία, θα μπορούσαν να μοντελοποιηθούν όσο και τα φυτά όσο και οι ακρίδες σαν δυο γενικές ομάδες που εκφράζουν τον κάθε πληθυσμό ξεχωριστά.

Τα μοντέλα που στηρίζονται σε ατομική μοντελοποίηση των οργανισμών χωρίζονται ως προς τον χρόνο και τον χώρο στις εξής 4 κατηγορίες [7]:

- Διακριτός Χώρος, Διακριτός Χρόνος
- Διακριτός Χώρος, Συνεχής Χρόνος
- Συνεχής Χώρος, Διακριτός Χρόνος
- Συνεχής Χώρος, Συνεχής Χρόνος

Διακριτός Χώρος, Διακριτός Χρόνος

Ο χώρος σε αυτό το είδος μοντέλου χωρίζεται σε ένα πίνακα $N \times M$ έχοντας τα όρια στις πλευρές, διασφαλίζοντας επίσης ότι σε κάθε κελί υπάρχει μόνο ένας οργανισμός κάθε χρονική στιγμή. Ο χρόνος είναι διακριτός γι αυτό σε κάθε βήμα χρόνου κάθε οργανισμός κάνει μια ενέργεια. Ο κάθε οργανισμός μπορεί να πεθάνει με κάποια προκαθορισμένη πιθανότητα P_θ και το ίδιο συμβαίνει με την αναπαραγωγή έχοντας μια άλλη πιθανότητα P_a . Όταν ένας οργανισμός προσπαθήσει να περάσει έξω από τα όρια είτε τοποθετείται τυχαία σε ένα σημείο (reflecting), και είναι ρεαλιστικό για οργανισμούς που ζουν σε ένα νησί, είτε εμφανίζεται στην ακριβώς απέναντι πλευρά (periodic), όπως το γνωστό παιχνίδι «Φιδάκι». Όλοι οι οργανισμοί κάνουν τα βήματα τους παράλληλα. Κάθε φάση αποτελείται από ένα βήμα χρόνου, δηλαδή πρώτα θα κάνουν όλοι κίνηση (πάνω, κάτω, δεξιά, αριστερά) και στο επόμενο χρονικό βήμα κάποια θα κάνουν αναπαραγωγή, κάποια θα πεθάνουν και κάποια τίποτα. Αυτό συμβαίνει αφού υπάρχει πιθανότητα σύγκρουσης σε κάποιο σημείο, όπου εφαρμόζονται κανόνες σύγκρουσης, αν δυο γεγονότα συμβούν ταυτόχρονα, τότε ένα από αυτά δεν κάνει τίποτα απολύτως. Αν προσομοιώνουμε φυτά προφανώς η φάση της κίνησης αγνοείτε. Κάθε απόγονος γίνεται μετά από κάποια χρονικά βήματα ενήλικας.

Διακριτός Χώρος, Συνεχής Χρόνος

Οι περιορισμοί του χώρου είναι οι ίδιοι και με το προηγούμενο είδος αφού ο χώρος είναι και εδώ διακριτός. Αυτό που αλλάζει είναι οι φάσεις τις ενηλικίωση, κίνηση, αναπαραγωγή και θνησιμότητα που επηρεάζονται άμεσα από τον χρόνο. Κάθε οργανισμός προχωρά σε μια ενεργεία από αυτές με ένα αντίστοιχο ξεχωριστό ρυθμό για κάθε φάση. Η φάση της ενηλικίωσης μπορεί μερικές φορές να γίνετε και απευθείας, θεωρώντας κάθε νέο απόγονο ενήλικο και έτοιμο για την φάση της αναπαραγωγής.

Συνεχής Χώρος, Διακριτός Χρόνος

Ο χώρος σε αυτό το είδος του μοντέλου είναι Συνεχής συνεπώς μοντελοποιείτε με ένα συνεχή χώρο μεγέθους $N \times M$ με χωρικά όρια. Αφού ο χρόνος είναι διακριτός και σε αυτό το μοντέλο όλα κάνουν μαζί τις ενέργειες σε κάθε βήμα χρόνου και πεθαίνουν ή αναπαράγονται με κάποια πιθανότητα P_θ και P_a . Κάθε νέος απόγονος εμφανίζεται ανάλογα με κάποια ακτίνα απόστασης μακριά από τους γονείς του χωρίς να ξεπερνούν τα όρια του χώρου, εξαρτώμενο συνήθως από τον αριθμό των οργανισμών μέσα σε εκείνο τον κύκλο. Κίνηση δεν έχει μόνο 4 κατευθύνσεις αλλά ένα κύκλο απόστασης που μπορούν να κινηθούν και μπορεί να γίνει παράλληλα με την εφαρμογή των κανόνων θνησιμότητας και αναπαραγωγής στα άλλα. Δεν μπορεί να επηρεαστεί το ίδιο σημείο ταυτόχρονα από διαφορετικά γεγονότα και να υπάρχει σύγκρουση, καθώς η θέση του οργανισμού στο χάρτη αποθηκεύετε στο κάθε οργανισμό ξεχωριστά.

Συνεχής Χώρος, Συνεχής Χρόνος

Ο χώρος συμπεριφέρεται όπως το προηγούμενο είδος, δηλαδή είναι Συνεχής συνεπώς μοντελοποιείτε με ένα συνεχή χώρο μεγέθους $N \times M$ με χωρικά όρια. Η αναπαραγωγή, θνησιμότητα και η κίνηση γίνετε με ένα αντίστοιχο ρυθμό στο κάθε οργανισμό ξεχωριστά. Δεν υπάρχουν χρονικά βήματα και όλα μπορούν να γίνουν παράλληλα. Ο επεξεργαστής επιλεγεί τυχαία ένα σύνολο από οργανισμούς και ελέγχοντας ένα-ένα αν πέθανε ή έκανε αναπαραγωγή. Αν έκανε αναπαραγωγή τοποθετεί τυχαία τον απόγονο του, αφού μπορεί παράλληλα να έκανε απόγονο ένας άλλος σε μια άλλη απόσταση έξω από το σύνολο που επέλεξε.

2.3 Έλεγχος Συστημάτων

Τα τελευταία 120 χρονιά η τεχνολογία έκανε την επανάσταση, μιας και οι υπολογιστές είναι πλέον μέρος της ζωής μας. Στον αιώνα της πληροφορίας, όπως ονομάζεται ο 21^{ος} αιώνας, οι πληροφορίες στοιχίζουν περισσότερο και από χρυσάφι. Καθημερινά χρησιμοποιούμε την τεχνολογία για να εργαστούμε, να τρικούμε, να επικοινωνήσουμε και να διασκεδάσουμε. Αυτό δημιουργεί την ανάγκη στο επίπεδο του λογισμικού και του υλικού να παρέχουν αξιοπιστία στον χρηστή. Λάθη όπως αυτά μπορούν να αποβούν μοιραία όπως το Ariane-5, το οποίο συντρίφθηκε 36 δευτερόλεπτα μετά την εκτόξευση, του οποίου το λάθος ήταν λόγο μιας μετατροπής δεκαδικού αριθμού από 64 bit σε 16 bit ακέραιο αριθμό. Οι κλασικές μέθοδοι για επαλήθευση κώδικα ήταν ανέκαθεν, η εξέταση του κώδικα από τρίτο άτομο, που αν και μπορεί να αναγνωρίσει ένα μεγάλο φάσμα λαθών δεν μπορεί να αναγνωρίσει περίπλοκα λογικά λάθη, και το Testing το οποίο καταναλώνει αρκετούς πόρους και πάλι δεν αναγνωρίζονται όλα τα λάθη. Λόγω αυτών των αναγκών αναπτυχτήκαν οι Τυπικοί Μέθοδοι. Στόχος των Τυπικών Μεθόδων είναι η απόδειξη της ορθότητας του μοντέλου ενός συστήματος με εφαρμοσμένα μαθηματικά. Μπορούν να προσφέρουν την δυνατότητα επαλήθευσης από την φάση του σχεδιασμού έτσι ώστε να μπορεί να ξανασχεδιαστεί σωστά πριν την υλοποίηση, κερδίζοντας έτσι χρόνο και μειώνοντας το κόστος. Παρέχουν με μαθηματική ακρίβεια ψηλότερη εγγύηση της ορθότητας από τις κλασικές μεθόδους. Τυπικές μέθοδοι είναι οι παραγωγικές μέθοδοι, που εφαρμόζονται αν το σύστημα έχει την μορφή μαθηματικής θεωρίας και δίνουν μια λογική απόδειξη ότι ισχύει μια ιδιότητα, οι προσομοιώσεις και πειράματα, που εφαρμόζονται σε εκτελέσιμα συστήματα και ελέγχουν αν ισχύει σε κάθε εκτέλεση του προγράμματος μια απαίτηση, και ο μοντελοέλεγχος.

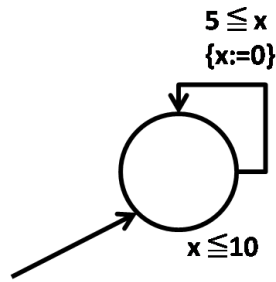
2.4 Θεωρία Αυτομάτων

Η Θεωρία Αυτομάτων είναι η μελέτη αφηρημένων μηχανών ή αυτόματα, όπως ονομάζονται αλλιώς, και των υπολογιστικών προβλημάτων που μπορούν να λάθουν με αυτά. Κάθε ένα μπορεί να εκφράσει μια τυπική γλωσσά, δηλαδή να απαντήσει θετικά μόλις αναγνωρίσει ένα μοτίβο. Τα αυτόματα αποτελούνται από κορυφές που αντιπροσωπεύουν καταστάσεις, τις ακμές που αντιπροσωπεύουν οι μεταβάσεις από την μια κατάσταση στην άλλη και τα βάρη που αντιπροσωπεύουν τις ενέργειες ή οι συνθήκες που προκαλούν την μετάβαση.

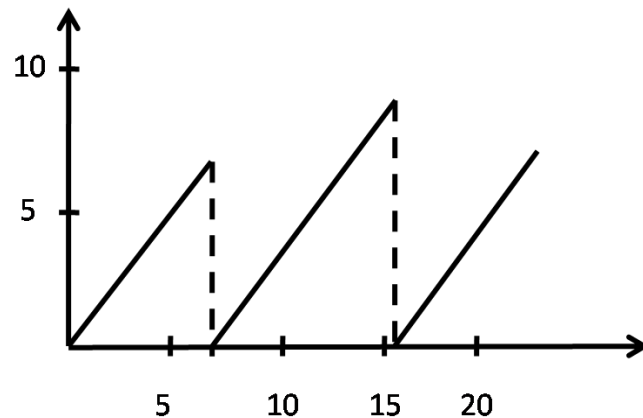
Μερικές φορές θέλουμε να εκφράσουμε στις μηχανές αυτές την έννοια του χρόνου. Ο χρόνος αυτός είναι συνεχής και μοντελοποιείται με αριθμούς. Μπορεί να συμβεί ένα γεγονός οποιαδήποτε χρονική στιγμή και ο χρόνος μεταξύ δυο γεγονότων μπορεί να είναι αυθαίρετα μικρός. Ένα αυτόματο που εκφράζει τον χρόνο ονομάζεται χρονικό αυτόματο. Τα χρονικά αυτόματα όπως και τα αλλα αυτόματα αποτελούνται από καταστάσεις και ακμές. Οι μεταβάσεις δεν μεταβάλουν τον χρόνο. Είναι εφοδιασμένο με ρολόγια τα οποία παίρνουν πραγματικές τιμές και αυξάνονται συγχρονισμένα με αρχική τιμή 0. Οι συνθήκες στις μεταβάσεις ελέγχονται από περιορισμούς στις τιμές των ρολογιών και ονομάζονται φρουροί των μεταβάσεων, οι οποίοι ελέγχουν αν θα γίνει μια μετάβαση ή όχι. Οι συνθήκες καταστάσεων εξαναγκάζουν την εκτέλεση μιας μετάβασης. Οι τιμές μπορούν να μηδενιστούν κατά την εκτέλεση μιας μετάβασης. Ένα χρονικό αυτόματο είναι μια πλειάδα $(C, L, l_0, \text{Label}, \text{invar}, E, \text{clocks}, \text{guard})$ όπου:

- C είναι το σύνολο ρολογιών
- L είναι ένα σύνολο από καταστάσεις με αρχική κατάσταση την l_0
- Label είναι μια συνάρτηση που αποδίδει σε κάθε κατάσταση ένα σύνολο από ατομικές προτάσεις (που ικανοποιούνται στην κατάσταση)
- Invar είναι μια συνάρτηση που αποδίδει σε κάθε κατάσταση μια συνθήκη κατάστασης
- E είναι ένα σύνολο από ακμές ανάμεσα στις καταστάσεις
- Clocks είναι μια συνάρτηση που αποδίδει σε κάθε ακμή το σύνολο των ρολογιών που πρέπει να μηδενιστούν.
- Guard είναι μια συνάρτηση που αποδίδει σε κάθε ακμή το φρουρό της ακμής

Ένα παράδειγμα χρονικού αυτομάτου είναι το Σχήμα 2.2 που δείχνει ένα αυτόματο που έχει κάτω φράγμα και συνθήκη κατάστασης. Ο χρόνος ξεκινά και μπορεί να πάρει την μετάβαση μονό όταν είναι μεταξύ 5 και 10 χρονικά διαστήματα. Το κατώτατο όριο το καθορίζει ο φρουρός της μετάβασης και το ανώτατο η συνθήκη της κατάστασης. Ο χρόνος κάθε φορά μηδενίζεται. Η γραφική παράσταση στο Σχήμα 2.3 δείχνει μια πιθανή εκτέλεση



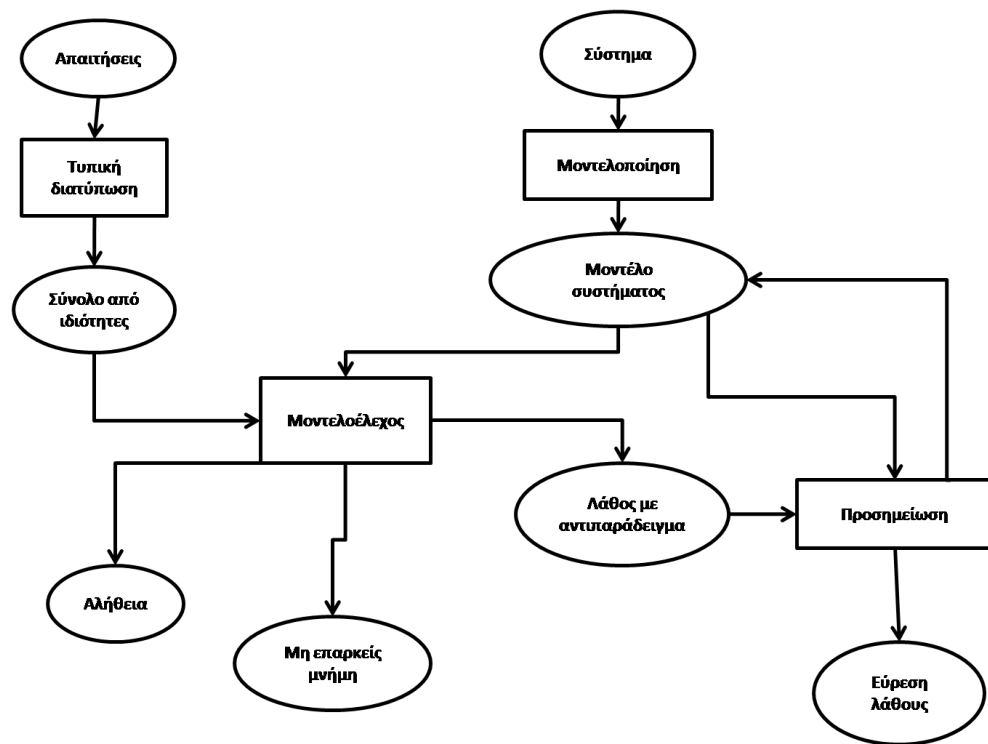
Σχήμα 2.2 Παράδειγμα Χρονικού Αυτομάτου



Σχήμα 2.3 Μια πιθανή εκτέλεση του Χρονικού Αυτομάτου στο Σχήμα 2.2

2.5 Μοντελοέλεγχος

Ο Μοντελοέλεγχος είναι ο έλεγχος σε ένα πεπερασμένο μοντέλο συμπεριφοράς και απαντά αν κάποια ιδιότητα ισχύει στις καταστάσεις του συστήματος. Ιδιότητες όπως για παράδειγμα αν υπάρχουν αδιέξοδα, αν δεδομένου κάποιας αρχικής κατάστασης μπορεί να υπάρξει σε κάποια άλλη κατάσταση ή αν απαντά σωστά το σύστημα και σε κάποιο χρονικό περιθώριο. Για να μπορεί να γίνει μοντελοέλεγχος πρέπει να γίνει ξεκάθαρη δήλωση των ιδιοτήτων που έχουν σκοπό να ελεγχτούν και αυτό γίνεται μέσω μιας χρονικής λογικής όπως οι CTL και PLTL. Μπορεί να εφαρμοστεί σε πολλά συστήματα υλικού και λογισμικού και η απάντηση είναι μαθηματικά αποδεδειγμένη σωστή, αφού μελετά όλες τις δυνατές εκτελέσεις και απαντά με αντιπαράδειγμα για μη-ικανοποίηση. Δυστυχώς για να γίνει τέτοια μελέτη πρέπει να γίνει σωστή κατασκευή των μοντέλων. Υπάρχουν αρκετά εργαλεία Μοντελοέλεγχου τα οποία θα μελετήσουμε μερικά από αυτά στο Κεφάλαιο 3.



Σχήμα 2.1 Το διάγραμμα εφαρμογής του Μοντελοέλεγχου

2.5.1 Χρονικές και τροπικές λογικές

Είναι οι γλώσσες που επινοήθηκαν για την διατύπωση ιδιοτήτων που αφορούν συμπεριφορά που μεταβάλετε με την πάροδο του χρόνου, κάτι που δεν μπορεί να εκφράσει ούτε η Κατηγορηματική Λογική ούτε ο προτασιακός λογισμός. Οι τροπικές λογικές αναπτύχθηκαν με σκοπό την μελέτη των τρόπων της αληθείας και κατηγορία τους είναι οι χρονικές λογικές οι οποίες οι ισχυρισμοί τους μεταβάλλονται στον χρόνο. Χρησιμοποιείτε για την προδιαγραφή και την επαλήθευση συστημάτων αλληλεπίδρασης. Αποτελείτε από τους παρακάτω χρονικούς τελεστές.

- $\langle \rangle \varphi$ ή $F \varphi$: Το φ θα συμβεί σε κάποια μελλοντική στιγμή
- $[] \varphi$ ή $G \varphi$: Το φ θα συμβεί σε κάθε μελλοντική στιγμή

Το φ είναι ατομική πρόταση που είναι και το βασικό στοιχείο της χρονικής λογικής. Οι ατομικές προτάσεις είναι λογικές εκφράσεις και αντιπροσωπεύουν μεταβλητές, σταθερές και κατηγορηματικά σύμβολα.

2.5.2 Προτασιακή Γραμμική Χρονική Λογική

Η προτασιακή Γραμμική Χρονική λογική (PLTL) μπορεί να έκφραση ιδιότητες όπως της ασφάλειας (τίποτα κακό δεν θα συμβεί ποτέ), της ζωτικότητας (κάτι κάλο κάποτε θα συμβεί) και της ανταπόκρισης (κάθε αίτηση θα εξυπηρετηθεί) και ορίζεται ως εξής:

$$\Phi ::= p \mid \neg \Phi \mid \Phi \vee \Psi \mid X \Phi \mid \Phi \cup \Psi$$

- Η p είναι ατομική πρόταση και είναι κάποια ιδιότητα
- Αν η Φ και η Ψ είναι ιδιότητες τότε και η $\neg \Phi$, $\Phi \vee \Psi$ είναι ιδιότητες
- Αν η Φ είναι ιδιότητα τότε και η $X \Phi$ είναι ιδιότητα, και δηλώνει ότι η Φ θα ισχύει ακριβώς μετά
- Αν η Φ και η Ψ είναι ιδιότητες τότε και η $\Phi \cup \Psi$ είναι ιδιότητες

Έχει τους παρακάτω παραγόμενους τελεστές:

- $\Phi \wedge \Psi \equiv \neg (\neg \Phi \vee \neg \Psi)$
- $\Phi \rightarrow \Psi \equiv \neg \Phi \vee \Psi$
- $\Phi \leftrightarrow \Psi \equiv (\Phi \rightarrow \Psi) \wedge (\Psi \rightarrow \Phi)$
- $\text{True} \equiv \neg \Phi \vee \Phi$
- $\text{False} \equiv \neg \text{True}$
- $F \Phi \equiv \text{True} \cup \Phi$ (κάποτε θα συμβεί)
- $G \Phi \equiv \neg F \neg \Phi$ (θα συμβαίνει πάντα)

2.5.3 Προτασιακή Διακλαδωμένη Χρονική Λογική

Η Διακλαδωμένη Χρονική Λογική (CTL) μας δίνει την δυνατότητα να εκφράσουμε διαφορετική κατηγορία ιδιοτήτων όπως η δυνατότητα να συμβεί κάτι και ορίζεται ως εξής:

$\Phi, \Psi ::= p \mid \neg \Phi \mid \Phi \vee \Psi \mid A \phi \mid E \phi$

$\phi ::= X \Phi \mid \Phi U \Psi \mid F \Phi \mid G \Phi$

- Η p είναι ατομική πρόταση και είναι ιδιότητα κατάστασης
- Αν η Φ και η Ψ είναι ιδιότητες κατάστασης τότε και η $\neg \Phi$, $\Phi \vee \Psi$ είναι ιδιότητες κατάστασης
- Αν η ϕ είναι ιδιότητα εκτέλεσης τότε, οι $A \phi$ και $E \phi$ είναι ιδιότητες κατάστασης,
- Αν η Φ και η Ψ είναι ιδιότητες τότε και οι $X \Phi$ και $\Phi U \Psi$ είναι ιδιότητες εκτέλεσης

Ο τελεστής A δηλώνει ότι η ιδιότητα ϕ ικανοποιείτε σε όλες τις εκτελέσεις του μοντέλου και ο τελεστής E δηλώνει υπάρχει μια εκτέλεση που ικανοποιεί την ιδιότητα ϕ .

Έχει τους παρακάτω παραγόμενους τελεστές:

- $F \Phi \equiv \text{True} U \Phi$
- $G \Phi \equiv \neg F \neg \Phi$
- $E F \Phi \equiv E(\text{True} U \Phi)$ είναι δυνατό να συμβεί
- $AG \Phi \equiv \neg EF \neg \Phi$ σταθερά συμβαίνει
- $A F \Phi \equiv A(\text{True} U \Phi)$ αναπόφευκτα θα συμβεί
- $EG \Phi \equiv \neg A F \neg \Phi$ δυνατόν πάντα να συμβεί

Η κάθε μια από τις 2 λογικές έχει δική της εκφραστικότητα. Υπάρχουν ιδιότητες που μπορεί μόνο η μια από τις δυο μπορεί να εκφράσει και όχι η άλλη όπως για παράδειγμα στην CTL $EFAGg$ που σημαίνει υπάρχει μονοπάτι που κάθε μονοπάτι μετά από εκείνο πάντα θα ισχύει το g ενώ στη PLTL δεν μπορώ να εκφράσω την ίδια ιδιότητα.

2.5.4 Στατιστικός Μοντελοέλεγχος

Ένα μεγάλο πρόβλημα που αντιμετωπίζει ο μοντελοέλεγχος είναι η εκθετική αύξηση των καταστάσεων και τον κάνει ακατόρθωτο να εκπληρωθεί, γι αυτό μπορεί να

χρησιμοποιηθεί μόνο σε μικρά μοντέλα, την λύση σε αυτό το πρόβλημα ήρθε να δώσει ο Στατικός Μοντελοέλεγχος. Ο Στατιστικός Μοντελοέλεγχος επιτυγχάνετε με το να τρέξει ένα μεγάλο αριθμό προσομοιώσεων παίρνοντας αρκετά τυχαία μονοπάτια και αρκετές πιθανές τιμές. Για κάθε πιθανή τιμή κάνει μια εκτίμηση ποσό βάρος θα δώσει σε εκείνη την τιμή με βάση την πιθανότητα να συμβεί εκείνο το μονοπάτι και εξάγει μια τελική τιμή. Σε αντίθεση με τον κανονικό μοντελοέλεχο βέβαια, που σου απαντά ξεκάθαρα, η απάντηση που θα σου εξάγει δεν είναι απαραίτητα σωστή αλλά έχει μια αποδεκτή απόκλιση από την πραγματική απάντηση. Η απόκλιση αυτή έχει να κάνει με το μέγεθος των εξαγόμενων αποτελεσμάτων μιας και όσα περισσότερα έχουμε στην διάθεση μας τόσο πιο κοντά στον πραγματικό αριθμό θα είναι.

Κεφάλαιο 3

Μελέτη Εργαλείων Μοντελοέλεγχου

3.1 Περιγραφή Parasitic Varroa mite	19
3.2 Μελέτη εργαλείου PRISM	20
3.2.1 Περιγραφή Εργαλείου PRISM	20
3.2.2 Μοντελοποίηση Parasitic Varroa mite	21
3.2.3 Αποτελέσματα	23
3.2.4 Συμπεράσματα	23
3.3 Μελέτη εργαλείου UPPAAL	24
3.3.1 Περιγραφή Εργαλείου UPPAAL SMC	24
3.3.2 Μοντελοποίηση Parasitic Varroa mite	25
3.3.3 Μοντελοποίηση Dynamic Template στο UPPAAL SMC	27
3.3.4 Αποτελέσματα	28
3.3.5 Συμπεράσματα	30
3.4 Επιλογή εργαλείου UPPAAL SMC	30

3.1 Περιγραφή Parasitic Varroa mite

Ο Parasitic Varroa mite είναι ένας πληθυσμός από τερμίτες που κινούνται σε ένα δισδιάστατο χώρο και κάθε άτομο τους αναπαράγεται αν δεν υπάρχει άλλο άτομο εκεί που βρίσκεται. Εμείς θα αναλύσουμε μια ελάχιστη λίγο πιο περιπλοκή, με παραπάνω ελέγχους, εκδοχή του πληθυσμού αυτού. Ο πληθυσμός αυτός ξεκινά και μπορεί να κινείται στο επίπεδο που του θέσαμε. Όταν συναντήσει ένα άλλο άτομο του ίδιου είδους, βρίσκονται δηλαδή στο ίδιο κελί και τα 2, προχωρούν στην αναπαραγωγή. Με κάποια πιθανότητα κάνουν ένα απόγονο και με την υπόλοιπη κάνουν δυο απογόνους. Οι απογόνοι θεωρούνται κανονικά μέλη του πληθυσμού και είναι ικανοί για μετακίνηση και αναπαραγωγή απευθείας μόλις γεννηθούν. Ο πληθυσμός αυτός θα χρησιμοποιηθεί στη μοντελοποίηση των 2 εργαλείων PRISM και UPPAAL, και στο

UPPAAL θα χρησιμοποιηθεί και για τα 2 είδη μοντελοποίησης που θα παρουσιαστούν, μοντελοποίηση με μέγιστο αριθμό και μοντελοποίηση με δυναμική αύξηση του πληθυσμού.

3.2 Μελέτη εργαλείου PRISM

3.2.1 Περιγραφή Εργαλείου PRISM

Το Prism είναι ένα εργαλείο μοντελοέλεγχου [17] και χρησιμοποιείται για την μοντελοποίηση και ανάλυση συστημάτων τα οποία έχουν μεταβάσεις βασισμένες στις πιθανότητες. Χρησιμοποιείται σε πολλούς τομείς όπως έλεγχος αλγορίθμων, θεωρίες παιγνίων, έλεγχος συστημάτων για απόδοση και εμπιστοσύνη, Βιολογία και Ασφάλεια. Υποστηρίζει τροπικές γλώσσες όπως PCTL, CSL, LTL και PCTL*.

Μπορεί να κτίσει και να λύσει διάφορα μοντέλα όπως

- Αλυσίδες Markov διακριτού χρόνου (DTMCs)
- Αλυσίδες Markov συνεχούς χρόνου (CTMCs)
- Διεργασίες αποφάσεων Markov (MDPs)
- Πιθανοτικά αυτόματα (PA)
- Πιθανοτικά χρονικά αυτόματα (PTAs)

καθώς και επεκτάσεις αυτών των μοντέλων με κόστη και οφέλη.

Η γλωσσά του PRISM βασίζεται στις καταστάσεις και τους φύλακες που καθορίζουν αν θα γίνει η ενέργεια ή όχι. Ένα μοντέλο στο PRISM αποτελείται από γενικές μεταβλητές οι οποίες τροποποιούνται και διαβάζονται από ένα σύνολο από υπομονάδες, αλληλεπιδρώντας έτσι μεταξύ τους. Κάθε υπομονάδα έχει τοπικές μεταβλητές που μπορούν να μεταβάλλονται μόνο από τον εαυτό τους αλλά διαβάζονται με από όλους. Οι εντολές με φύλακες αποτελούν την συμπεριφορά των υπομονάδων και είναι της μορφής.

`[] guard -> prob_1:update_1+...+prob_n:update_n;`

Μπορούμε να τα συγχρονίσουμε βάζοντας μια λέξη πχ tick μέσα στα [tick] στις αντίστοιχες υπομονάδες που θέλω. Ο guard είναι ένα κατηγορημα το απαντά αν μπορεί

να εκτελεστεί με κάποια πιθανότητα κάποια από τις ενέργειες. Τα $prob_i$ είναι μια τιμή που αντιπροσωπεύει μια πιθανότητα $(0,1]$ και τα $update_i$ είναι αναθέσεις που δείχνουν την τιμή που θα πάρει κάποια μεταβλητή μετά την εκτέλεση της ενέργειας αυτής πχ η p θα γίνει 1 ($p'=1$). Αν σε μια υπομονάδα ισχύουν πολλές εντολές τότε παίρνει την πρώτη από αυτές. Ο μοντελοέλεγχος γίνεται με τον κλασικό τρόπο

3.2.2 Μοντελοποίηση Parasitic Varroa mite

Η αρχική μοντελοποίηση του Parasitic Varroa mite έγινε με αρχικό πληθυσμό 2 σε ένα πίνακα 3×3 . Όπως βλέπουμε στο Σχήμα 3.1, Σχήμα 3.2 και Σχήμα 3.3 υπάρχουν 3 είδη module, το master, τα αρχικά mites και αυτά που γεννιούνται μετά. Το master δημιουργήθηκε για να πάρει τη θέση του ρολογιού έτσι συντονίζει κάθε φάση των τερμιτών να γίνεται ταυτόχρονα. Δηλαδή πρώτα είναι όλα στην φάση της κίνησης, μετά κάνουν όλα μια κίνηση και στη συνέχεια περνούν στη φάση της αναπαραγωγής, όπου δημιουργούνται οι πιθανοί απόγονοι και τέλος επαναλαμβάνει τον κύκλο αυτό. Όπως φαίνεται στο Σχήμα 3.1 το Part A είναι υπεύθυνο για να λέει στα mites ότι μπορούν να κινηθούν. Το Part B αλλάζει την κατάσταση των mites έτσι ώστε είναι έτοιμα για έλεγχο αναπαραγωγής. Το Part C κάνει τον έλεγχο αν είναι στο ίδιο κελί δυο mites και κάνει την αναπαραγωγή. Τέλος το Part D είναι για να επιτρέπει να μπορούν να προχωρήσουν όσα mites δεν έχουν να κάνουν αναπαραγωγή. Κάθε module από mite μπορεί να κινείται με ίση πιθανότητα πάνω, κάτω, δεξιά, αριστερά έχοντας πάνω του μεταβλητές αν μετακινήθηκε, αν έκανε αναπαραγωγή και είναι γεννημένο. Ο μέγιστος αριθμός πληθυσμού πρέπει να προκαθοριστεί και να γίνουν οι απαραίτητοι έλεγχοι κάθε Starting mite ή mite με κάθε άλλο. Το Starting mite 0 ξεκινά από την θέση 0, 0, το Starting mite 1 από την θέση N-1, N-1 και όλα τα υπόλοιπα από την θέση -1, -1. Στην φάση της αναπαραγωγής κάθε γεννημένο mite ελέγχει τη θέση του με κάθε άλλο γεννημένο και αν βρει κάποιο που έχουν την ίδια θέση προχωράνε στην αναπαραγωγή. Μόνο μια αναπαραγωγή μπορεί να γίνει κάθε φορά λόγω του ότι οι νέοι οργανισμοί γεννιούνται στο ίδιο σημείο και θα γινόταν λάθος αναπαραγωγή αφού θα είχαν την ίδια θέση με τους δυο γονείς καθώς επίσης αν υπήρχαν 3 ή 4 οργανισμοί στο ίδιο κελί θα έκαναν όλοι απογόνους με όλους. Κάθε νέος απόγονος παίρνει τη θέση του πατέρα του (βλέπε ΠΑΡΑΡΤΗΜΑ Α).

```

module master

[] move_ready_rep=2 & num_mites=2 & left_to_reproduce=0->
(move_ready_rep'=0)&(moved0'=0) & (moved1'=0 );
[] move_ready_rep=2 & num_mites=3 & left_to_reproduce=0->
(move_ready_rep'=0)&(moved0'=0) & (moved1'=0 ) & (moved2'=0 );
[] x0=x1&y0=y1&move_ready_rep=1-
>repr:(num_mites'=num_mites+1)&(Xparent'=x0)&(Yparent'=y0)
&(move_ready_rep'=2)&(left_to_reproduce'=left_to_reproduce+1)+
(1-
repr):(num_mites'=num_mites+2)&(Xparent'=x0)&(Yparent'=y0)&(move_ready_rep'=2)
&
(left_to_reproduce'=left_to_reproduce+2);
[] x0=x2&y0=y2&move_ready_rep=1->repr:(num_mites'=num_mites+1)&(Xparent'=x0)
&(Yparent'=y0)&(move_ready_rep'=2)&(left_to_reproduce'=left_to_reproduce+1)+
(1-
repr):(num_mites'=num_mites+2)&(Xparent'=x0)&(Yparent'=y0)&(move_ready_rep'=2)
& (left_to_reproduce'=left_to_reproduce+2);
[] ((x0!=x1|y0!=y1)& move_ready_rep=1 & live1=1 & num_mites=id2)-
>(move_ready_rep'=2);
[] ((x0!=x1|y0!=y1)&(x0!=x2|y0!=y2)&(x1!=x2|y1!=y2)& move_ready_rep=1 &
live2=1 & num_mites=id3)->(move_ready_rep'=2);
endmodule

```

Σχήμα 3.1: To Master Module στην γλώσσα του εργαλείου Prism

```

module s0
x0:[0..N] init 0;
y0:[0..N] init 0;
live0 : int init 1;
[] num_mites>id0 &move_ready_rep=0&moved0=0->
0.25:(x0'=min(x0+1,N))&(moved0'=1)//aristera
+0.25:(y0'=min(y0+1,N))&(moved0'=1)//pano
+0.25:(y0'=max(y0-1,0))&(moved0'=1)//dekisa
+0.25:(x0'=max(x0-1,0))&(moved0'=1)//kato
endmodule

```

Σχήμα 3.2: To Starting Mite Module στην γλώσσα του εργαλείου Prism

```

module s2
live2 : int init 0;
[] num_mites>id2 & live2=0 ->(live2'=1)&(x2'=Xparent)&
(y2'=Yparent)&(moved2'=0)&(left_to_reproduce'=left_to_reproduce-1);
[] num_mites>id2 & live2=1 &move_ready_rep=0& moved2=0->
0.25:(x2'=min(x2+1,N))&(moved2'=1)//aristera
+0.25:(y2'=min(y2+1,N))&(moved2'=1)//pano

```

```

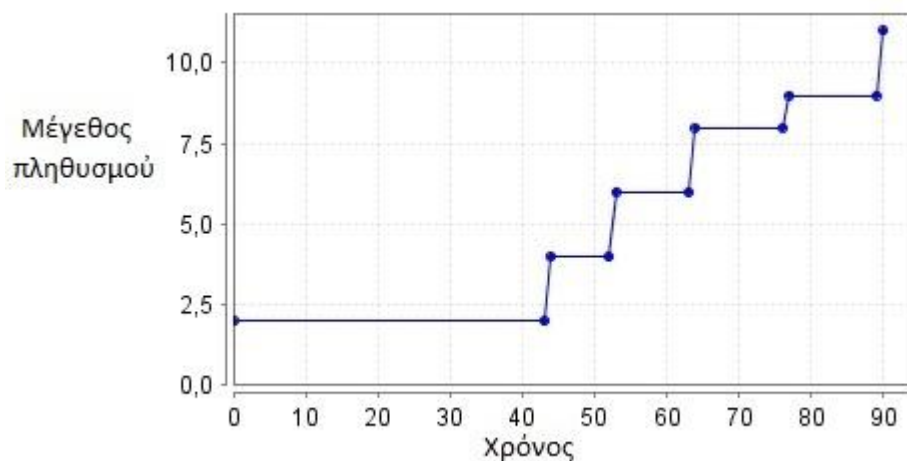
+0.25: (y2'=max(y2-1,0)) & (moved2'=1) //dekisa
+0.25: (x2'=max(x2-1,0)) & (moved2'=1) ; //kato
endmodule
module s3 = s2 [ x2=x3, y2=y3, live2=live3, moved2=moved3, id2=id3] endmodule
module s4 = s2 [ x2=x4, y2=y4, live2=live4, moved2=moved4, id2=id4] endmodule
module s5 = s2 [ x2=x5, y2=y5, live2=live5, moved2=moved5 , id2=id5] endmodule
module s6 = s2 [ x2=x6, y2=y6, live2=live6, moved2=moved6 , id2=id6] endmodule
module s7 = s2 [ x2=x7, y2=y7, live2=live7, moved2=moved7 , id2=id7] endmodule
module s8 = s2 [ x2=x8, y2=y8, live2=live8, moved2=moved8 , id2=id8] endmodule

```

Σχήμα 3.3: Τα Mite Module που γεννιούνται στην ποριά στην γλώσσα του εργαλείου Prism

3.2.3 Αποτελέσματα

Τα αποτελέσματα όπως βλέπουμε και από το Σχήμα 3.4 είναι τα αναμενόμενα δηλαδή να έχει εκθετική αύξηση αφού αν δυο μέλη του πληθυσμού είναι στο ίδιο κελί κάνουν σίγουρα ένα απόγονο και με πιθανότητα 0.7 η οποία είναι αρκετά μεγάλη. Αφού μόνο 9 κελιά υπάρχουν είναι σχεδόν αδύνατο σε κάθε κίνηση να μην γίνει κάποια αναπαραγωγή. Ο χρόνος αντιπροσωπεύει το ποσό των εντολών που εκτελεστήκαν σε σειρά, αφού κάθε εντολή θεωρείτε ένα χρονικό διάστημα, εκτός οι εντολές που εκτελεστήκαν παράλληλα με την χρήση του συγχρονισμού και μέτρησαν σαν μια.



Σχήμα 3.4: Η αύξηση του πληθυσμού στο Parasitic Varroa mite σε χώρο 3x3 στον μοντελοελεκτή PRISM

3.2.4 Συμπεράσματα

Η ανάλυση των αποτελεσμάτων ήταν αρκετά ενθαρρυντική ως προς την ορθότητα του μοντέλου. Παρόλα αυτά η κατανόηση και έκφραση της γλώσσας ήταν αρκετά δύσκολη. Υπήρχαν αρκετοί περιορισμοί όπως το κάθε υποσύστημα μπορεί να επεξεργαστεί μόνο τις δίκες του μεταβλητές. Δεν υπήρχαν ατομικά ρολόγια αρά έπρεπε να γίνει ο συγχρονισμός με κανάλια επικοινωνίας και να προγραμματιστούν με τρόπο που δεν επέμβαινε το ένα στο άλλο.

3.3 Μελέτη εργαλείου UPPAAL

3.3.1 Περιγραφή Εργαλείου UPPAAL SMC

Το εργαλείο UPPAAL είναι ένα εργαλείο μοντελοέλεγχου, προσομοίωσης και επαλήθευσης συστημάτων πραγματικού χρόνου [18]. Είναι κατάλληλο για συστήματα που μπορούν να μοντελοποιηθούν σαν σύνολο από μη ντετερμινιστικές διεργασίες με περιορισμένο έλεγχο δομής, ρολόγια πραγματικών αριθμών και με επικοινωνίες μέσω καναλιών και κοινών μεταβλητών. Σου δίνει την δυνατότητα να γράψεις συναρτήσεις με τις βασικές αρχές προγραμματισμού μέσω μιας γλώσσας σαν την C. Αποτελείτε από την περιγραφική γλώσσά, τον προσομοιωτή και τον μοντελοελεκτή. Η περιγραφική γλώσσα είναι για να δημιουργείς τις σχέσεις μεταξύ των καταστάσεων και των αυτόματων καθώς και την αλληλεπίδραση τους. Ο προσομοιωτής σε βοηθά να μπορείς να τρέξεις ένα πιθανό σενάριο και να ελέγξεις μόνος τον τρόπο που δουλεύει το σύστημα σου. Ο μοντελοελεκτή μπορεί να απαντήσει σε ιδιότητες που του θέτονται και μπορεί με τα αποτελέσματα να εξάγει γραφικές παραστάσεις, και γίνετε Στατιστικός Μοντελοέλεγχος στα μοντέλα του.

Template: τα αυτόματα που καθορίζονται με ένα σετ από παραμέτρους των βασικών τύπων (int, char) τα οποία μπορούν να δοθούν και σαν παράμετροι στο αυτόματο αυτό.

Binary synchronization: τα κανάλια δηλώνονται σαν chan c. Σκοπό έχουν να συγχρονίσουν 2 αυτόματα να μετακινηθούν μαζί. Από την μια μεριά είναι το c! και από την άλλη το c? τα οποία συγχρονίζονται, επιλέγοντας τυχαία αν υπάρχουν πολλοί υποψήφιοι με c?.

Broadcast channels: είναι τα κανάλια τα οποία δηλώνονται σαν broadcast chan c. Από την μια μεριά είναι το c! σαν ρόλος αποστολέα και από την άλλη κάποιος αριθμός από c? τα οποία συγχρονίζονται όλα μαζί όταν λάβουν το μήνυμα. Το c! κάνει την κίνηση άσχετα αν υπάρχει παραλήπτης όπως το κανονικό broadcast.

Committed Locations: Όταν ένα σύστημα είναι σε committed locations ο χρόνος θεωρείται ότι δεν μεταβάλετε και η επόμενη πάντα κίνηση που θα πάρει το σύστημα είναι μια από το σύνολο όλων των αυτομάτων που βρίσκονται σε μια τέτοια κατάσταση.

Guard: Είναι ένας έλεγχος που μπαίνει σε μια μετακίνηση από την μια κατάσταση στην άλλη. Για να μπορεί να περάσει πρέπει να άπαντα στην συνθήκη του guard true.

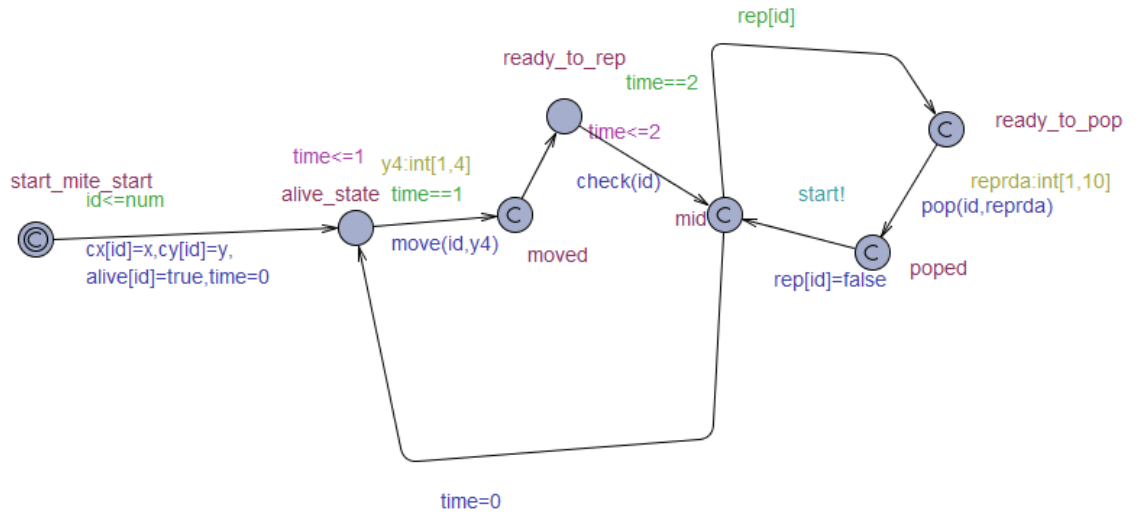
Invariant: Είναι μια συνθήκη που μπαίνει πάνω στις καταστάσεις και κανένα αυτόματο δεν μπορεί να είναι σε κατάσταση τις οποίας δεν εκπληρώνει την συνθήκη αυτή και ούτε μπορεί να μετακινηθεί σε αυτή. Αν πάει σε μια τέτοια κατάσταση ή είναι σε αυτή και δεν μπορεί αν φύγει με κάποια μετακίνηση από το σύνολο των μετακινήσεων που μπορεί να κάνει από αυτή την κατάσταση παρουσιάζετε deadlock.

Η γλώσσα των ιδιοτήτων (query language) είναι ένα απλοποιημένο είδος CTL και μπορεί να ελέγξει ιδιότητες όπως τις ζωτικότητα, ασφάλειας και ανταπόκρισης. Υπάρχουν έξτρα κανόνες που εισάγονται στο UPPAAL SMC για στοχαστική ανάλυση των χρονικών αυτομάτων και είναι:

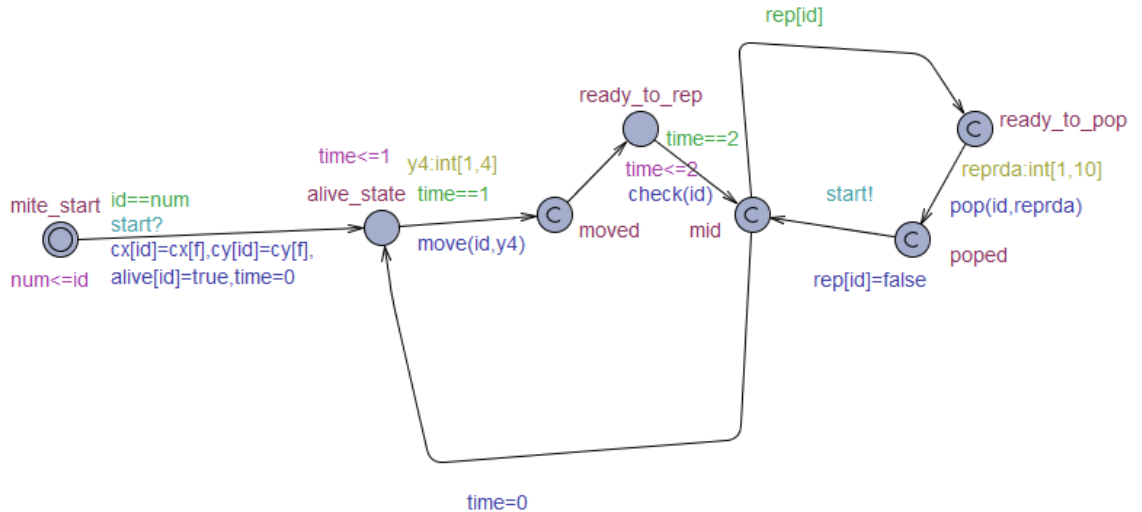
- **Simulate** μοντελοποιεί ένα αριθμό από εκτελέσεις απαντώντας στην αλλαγή μιας μεταβλητής στην πάροδο του χρόνου ($\text{simulate } N[\leq \text{bound}] \{ E1, \dots, Ek \}$)
- **Pr** είναι για την εκτίμηση πιθανότητας να συμβεί μια ιδιότητα σε κάποιο χρόνο ($\text{Pr } [\text{bound}] (\psi)$)
- **E** είναι η εκτίμηση των αναμενομένων μέγιστων ή ελάχιστων τιμών, λαμβάνοντας τα ρολόγια ($E[\text{bound}; N] (\text{min:expr})$ ή $E[\text{bound}; N] (\text{min:expr})$)

3.3.2 Μοντελοποίηση Parasitic Varroa mite

Το Parasitic Varroa mite μοντελοποιήθηκε με πιο αυτόνομη λογική στο UPPAAL, έχοντας μόνο τα 2 modules των mites χωρίς την ανάγκη για το Master Template αφού ο συγχρονισμός και τα ρολόγια υπάρχουν υλοποιημένα στο εργαλείο. Όπως βλέπουμε και στο Σχήμα 3.5 ξεκινούν τα αρχικά mites και αρχικοποιούνται με τις θέσεις που τους δίνονται κάνοντας έτσι τον κύκλο της ζωής τους. Στο Σχήμα 3.7 τα mites περιμένουν μέχρι να αυξηθεί ο πληθυσμός που σημαίνει έγινε αναπαραγωγή και περνούν την ίδια θέση που είχε ο πατέρας του για αρχικοποίηση (βλέπε ΠΑΡΑΡΤΗΜΑ Β).



Σχήμα 3.5: Το Starting Mite Template για τα Parasitic Varroa mite σε μοντελοποίηση στο UPPAAL SMC

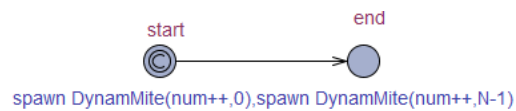


Σχήμα 3.6: Το Mite Template για τα Parasitic Varroa mite σε μοντελοποίηση στο UPPAAL SMC

Ο κύκλος ζωής των mites είναι πολύ απλός και είναι ο εξής: Ξεκινώντας από την κατάσταση `alive_state` όπου περιμένουν μέχρι να περάσει μια μονάδα χρόνου για όλα τα Template με σκοπό να συντονιστούν. Αυτό συμβαίνει λόγω του ότι στις προηγούμενες καταστάσεις δεν μετρούσε ο χρόνος (Committed states) αρά όλα έφτασαν εκεί χωρίς να περάσει ο χρόνος σε αυτά που ήταν ήδη σε αυτή την κατάσταση. Έπειτα επιλέγουν τυχαία μια κατεύθυνση και κινούνται όλα μαζί. Αναμένουν μια μονάδα χρόνου και μαζί ελέγχουν αν έχουν την ίδια θέση με κάποιο άλλο και προχωρούνε στην αύξηση του πληθυσμού, αν κάνουν αναπαραγωγή, αλλιώς επιστρέφουν στην αρχική κατάσταση `alive_state` και επαναλαμβάνουν τον κύκλο.

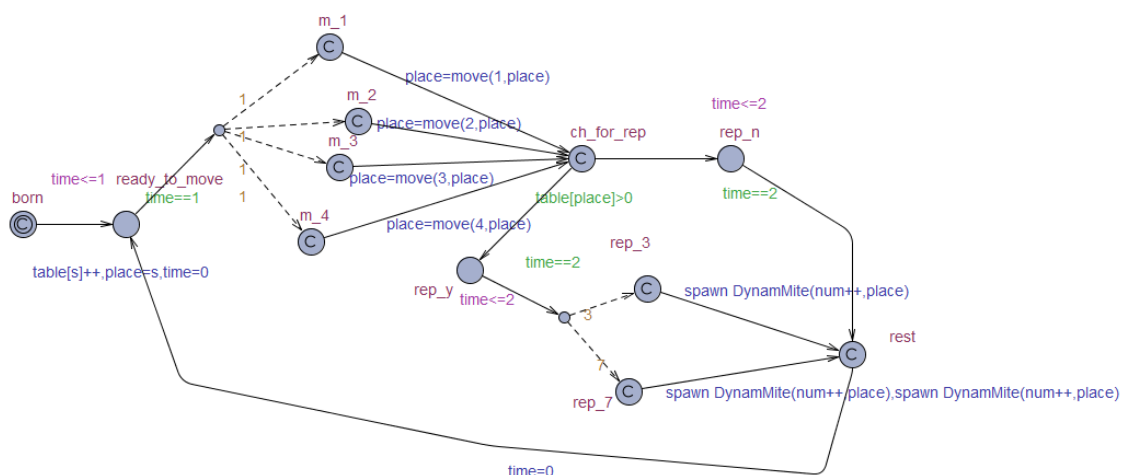
3.3.3 Μοντελοποίηση Dynamic Template στο UPPAAL SMC

Το μοντέλο αυτό έχει το Master Template για τη δημιουργία των αρχικών Mites και σε αντίθεση με τα άλλα 2 (3.2.2 και 3.3.2) χρειάζεται μόνο ένα Template από mite, το DynamMite Template, που μοντελοποιεί τη ζωή του συγκεκριμένου οργανισμού. Κατά την έναρξη της μοντελοποίησης υπάρχει μόνο το Master Template, όπως φαίνεται στο Σχήμα 3.7, και δημιουργεί τα άλλα DynamMite Template (βλέπε ΠΑΡΑΡΤΗΜΑ C).



Σχήμα 3.7: Το Master Template για τα Parasitic Varroa mite σε μοντελοποίηση με δυναμική δημιουργία Template στο UPPAAL SMC

Η μοντελοποίηση του δισδιάστατου χώρου έγινε σε ένα πίνακα με θέσεις και η σύγκριση για το αν 2 mites είναι στο ίδιο κελί ελέγχεται με τον αριθμό των mites που αποθηκεύεται στο συγκεκριμένο κελί. Όπως και στις προηγούμενες μοντελοποιήσεις (3.2.2 και 3.3.2) τα 2 αρχικά ξεκινούν στο 0, 0 και N-1, N-1.



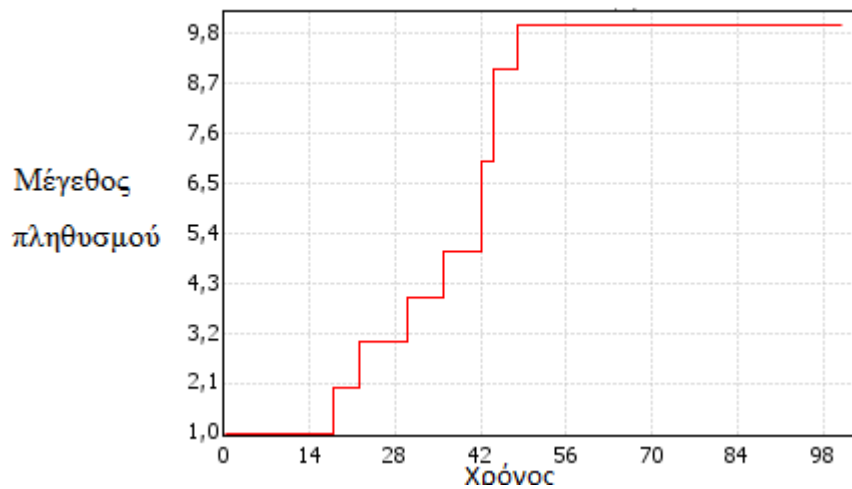
Σχήμα 3.8: Το DynamMite Template για τα Parasitic Varroa mite σε μοντελοποίηση με δυναμική δημιουργία Template στο UPPAAL SMC

Στο Σχήμα 3.8 τα mites γεννιούνται και περιμένουν να συγχρονιστούν όλα για να κάνουν την μετάβαση για την κίνηση τους. Εκεί επιλέγουν μιαν από τις 4 κατευθύνσεις για να κινηθούν και μετακινούνται αυξάνοντας τον αριθμό των τερμιτών στο νέο κελί που μετακινηθήκαν και μειώνοντας τον αριθμό στο αντίστοιχο παλιό κελί. Στην συνέχεια κάθε τερμίτης που βρίσκεται σε κελί το οποίο υπάρχουν και άλλοι τερμίτες προχωρά στην φάση της αναπαραγωγής. Γίνεται μετά η επιλογή με το Probabilistic Weight αν θα κάνουν 1 ή 2 απογόνους και προχωράνε στην δημιουργία τους με την εντολή `spawn DynamMite(num++,place)` δίνοντας τους τη δική τους θέση σαν την αρχική τους θέση του απόγονου.

3.3.4 Αποτελέσματα

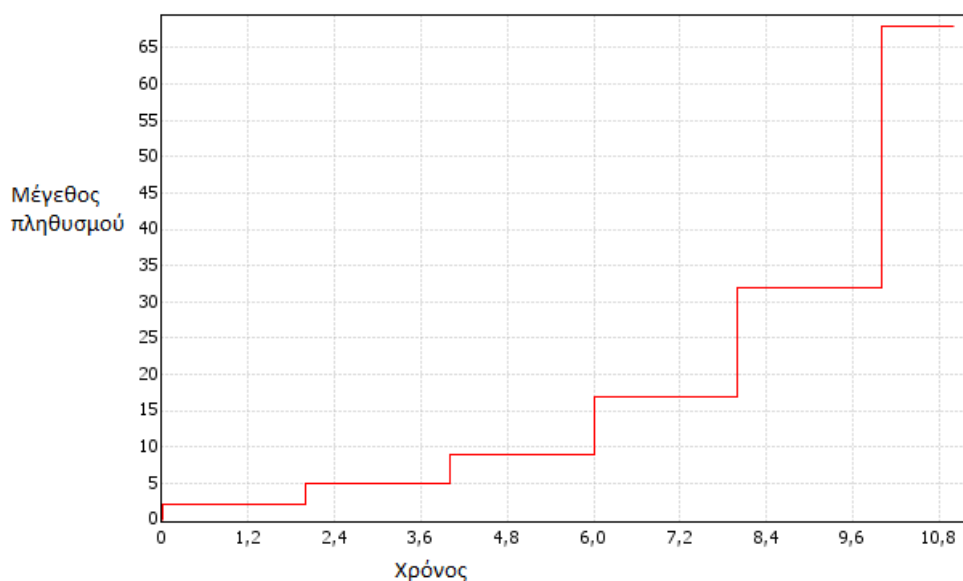
Στο Σχήμα 3.9 είναι η γραφική παράσταση του αριθμού του πληθυσμού του Parasitic Varroa σε σχέση με μονάδες χρόνου. Βλέπουμε ότι έχει την ίδιο ρυθμό, που ήταν αναμενόμενο, όπως και στο Σχήμα 3.4. Παρατηρούμε ότι η αύξηση του πληθυσμού γίνεται ακόμα πιο απότομη καθώς μεγαλώνει ο αριθμός λόγω του ότι υπάρχουν περισσότερα mites άρα περισσότερη πιθανότητα 2 από αυτά να είναι στο ίδιο κελί. Ο χρόνος εδώ δεν μετρίεται σαν εντολές αλλά σαν κύκλοι ζωής του πληθυσμού. Ένας κύκλος ζωής του πληθυσμού αποτελείται από 2 χρονικά διαστήματα τα οποία ονομάζονται μετακίνηση και έλεγχος για αναπαραγωγή. Έτσι αν θεωρήσουμε τα

χρονικά βήματα σαν δευτερόλεπτα τότε το ένα δευτερόλεπτο όλος ο πληθυσμός κινείται και το άλλο κάνουν όλοι ελέγχουν για αναπαραγωγή και όσοι μπορούν κάνουν



Σχήμα 3.9: Η γραφική παράσταση του αριθμού του πληθυσμού Parasitic Varroa σε σχέση με τον χρόνο για μοντελοποίησης στο UPPAAL χωρίς δυναμική δημιουργία Template

Στο Σχήμα 3.10 παρουσιάζεται η γραφική παράσταση του αριθμού του πληθυσμού του Parasitic Varroa σε σχέση με μονάδες χρόνου για δυναμική δημιουργία mite Template. Το πιο αξιοσημείωτο αυτής της γραφικής παράστασης είναι ότι λόγω του ότι ο πληθυσμός δεν έχει κάποιο όριο μπορεί να αυξηθεί με μεγαλύτερη κλήση η γραφική παράσταση. Για παράδειγμα όπως φαίνεται και στο Σχήμα 3.10 η αύξηση μεταξύ του χρονικού διαστήματος 9.6 και 10.8 είναι περίπου αύξηση 30 μονάδες πληθυσμού. Αρκετά μεγάλο αν αναλογιστούμε ότι ο αριθμός του πληθυσμού εκείνη τη χρονική στιγμή είναι μόλις 35. Ο χρόνος στο σχήμα αυτό έχει την ίδια έννοια με τον προηγούμενο που μετρίεται δηλαδή σαν κύκλοι ζωής του πληθυσμού. Ένας κύκλος ζωής του πληθυσμού αποτελείτε από 2 χρονικά διαστήματα τα οποία ονομάζονται μετακίνηση και έλεγχος για αναπαραγωγή.



Σχήμα 3.10: Η γραφική παράσταση του αριθμού του πληθυσμού Parasitic Varroa σε σχέση με τον χρόνο για μοντελοποίησης στο UPPAAL SMC με δυναμική δημιουργία Template

3.3.5 Συμπεράσματα

Αξιοποιώντας τα αποτελέσματα από τις μοντελοποιήσεις στο UPPAAL και UPPAAL SMC κατάλαβα ότι μπορούμε να μοντελοποιήσουμε ένα πληθυσμό με αρκετή ακρίβεια αναπαριστώντας την ορθή συμπεριφορά του. Οπτικά μπορώ να διακρίνουμε κάθε κατάσταση σε ποια φάση της ζωής του αντιστοιχεί και ποιές είναι όλες τις πιθανές καταστάσεις που μπορεί να κινηθεί. Έχω την δυνατότητα να εξάγω γραφικές παραστάσεις κάνοντας μοντελοέλεχο του μοντέλου με confidence interval $\varepsilon=5\%$. Επίσης μπορούμε να σχεδιάσουμε μοντέλα τα οποία δεν θέλεις να βάλεις περιορισμό για μέγιστο αριθμό πληθυσμού αλλά δημιουργώντας δυναμικά άλλα Template.

3.4 Επιλογή εργαλείου UPPAAL SMC

Το εργαλείο Prism παρέχει ένα μάλλον μη φιλικό περιβάλλον για το χρηστή αφού πρέπει να προγραμματίσεις κάθε κατάσταση και που είναι πιθανό να κινηθεί από αυτή. Δεν υπάρχουν ωστόσο καταστάσεις σαν οντότητες αλλά καθορίζονται η κάθε μια από τις τιμές που έχουν οι μεταβλητές του κάθε Module. Ο προγραμματισμός σε αυτό το εργαλείο είναι αρκετά δύσκολος αφού δεν υπάρχουν ρολόγια για συγχρονισμό, κάθε μοντέλο μπορεί να έχει μόνο μεταβλητές και τι μετάβαση μπορεί να κάνει χωρίς να

δικαιούται να μεταβάλει public μεταβλητές. Το μέγεθος του κώδικα για τη μοντελοποίηση ενός απλού πληθυσμού που κάνει απλές διεργασίες στον κύκλο ζωής του ήταν αρκετά μεγάλο και ο κώδικας ήταν αρκετά δυσανάγνωστος.

Το εργαλείο UPPAAL από την άλλη μεριά παρέχει ένα γραφικό περιβάλλον δημιουργίας του μοντέλου καθώς και υποστήριξη των βασικών λειτουργιών της γλώσσας προγραμματισμού C. Αρκετά πιο εύκολο να προσθέσεις νέους παράγοντες που επηρεάζουν το μοντέλο. Οι καταστάσεις είναι εμφανείς καθώς και οι μεταβάσεις τους και μπορούν να αντιστοιχηθούν άμεσα με πραγματικές καταστάσεις στον κύκλο ζωής των πληθυσμών. Έχει υλοποιημένα τα ρολόγια για να μπορείς να συγχρονίσεις τα Template να κινούνται παράλληλα σε ένα διάστημα χρόνου, προσομοιώνοντας έτσι το πραγματικό πέρασμα του χρόνου. Αυτό συμβαίνει διότι υπάρχουν οι ιδιότητες Committed και Urgent που απαγορεύουν το πέρασμα του χρόνου κατά τη διάρκεια που εκτελούνται, κάνοντας όλα τα μοντέλα να μετρούν χρόνο μαζί στις επιθυμητές καταστάσεις. Παρέχει τον συντονισμό κάποιων αυτομάτων, κάνοντας τα να κινούνται μαζί σε κάποιες μεταβάσεις, με την χρήση των καναλιών.

Η επέκταση του εργαλείου UPPAAL, UPPAAL SMC, παρέχει τις ίδιες δυνατότητες με το κανονικό εργαλείο αλλά παρέχει και στατιστικό έλεγχο, αφού είναι αποδεκτός για μοντελοποίηση βιολογικών συστημάτων και βέβαια αρκετά προτιμότερο για πιο μεγαλύτερα μοντέλα, καθώς και μετακινήσεις από καταστάσεις σε καταστάσεις με πιθανοτικές μεταβάσεις. Μπορεί να εξάγει Probability Density Distribution γραφικές παραστάσεις για κάποια μεταβλητή, με confidence interval και μοντελοέλεγχο. Τέλος μπορεί να δημιουργήσει δυναμικά Template αντίγραφα κάποιου Template και να τα καταστρέψει απλοποιώντας και κάνοντας πιο ρεαλιστική την μοντελοποίηση της ζωής ενός οργανισμού.

Μελετώντας τα συμπεράσματα για τα δυο εργαλεία είναι εμφανές ότι υπερισχύει το UPPAAL SMC λόγω του ότι είναι πιο εύχρηστο, έχει περισσότερες δυνατότητες μοντελοποίησης και Μοντελοέλεγχου που βοηθούν στην κατασκευή ενός σωστού και ρεαλιστικού μοντέλου, είναι επεκτάσιμο πιο εύκολα για μεγαλύτερες κλίμακες πληθυσμού καθώς και δυναμική ανάπτυξη του πληθυσμού.

Κεφάλαιο 4

Ο Μαυροπετρίτης και Ανάλυση των Δεδομένων

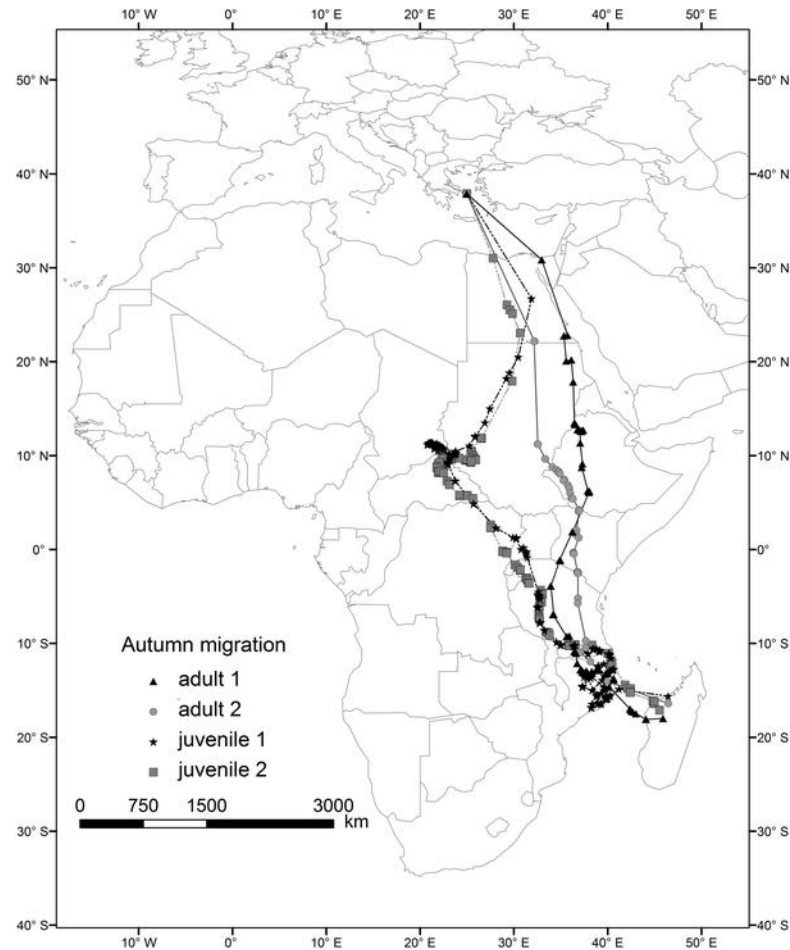
4.1 Μαυροπετρίτης	32
4.2 Ανάλυση των δεδομένων	34

4.1 Μαυροπετρίτης

Ο Μαυροπετρίτης είναι ένα είδος γερακιού και πηρέ το όνομα του από την Ελεονόρα Ντε Μπας πριγκίπισσα της Σαρδηνίας(1347-1404) η οποία αγαπούσε το συγκεκριμένο είδος [13]. Από τον Μάρτιο και κυρίως μέσα του Απριλίου ο Μαυροπετρίτης αναπαράγεται στην Βόρεια Αφρική, Κανάριους νήσους, Μεσογειακή Ευρώπη και κυρίως στο Αιγαίο Πέλαγος, και μεταναστεύει πλήρως στην Μαδαγασκάρη και τα γύρο νησιά κατά τον Οκτώβριο - Νοέμβριο. Τρέφεται από τον Απρίλιο έως τον Σεπτέμβριο με μεγάλα έντομα, τρωκτικά και λίγα μικρά πουλιά. Από τον Σεπτέμβριο μέχρι τον Νοέμβριο, και αυτός είναι λόγος που αργεί η μετανάστευση του, τρέφονται σχεδόν μόνο με μεταναστευτικά πουλιά που ταξιδεύουν από την Ευρώπη για την Αφρική, καθώς είναι και ο καιρός που εκκολάπτονται τα αυγά τους. Είναι το μοναδικό γεράκι με αυτή την συμπεριφορά [13]. Τα νεαρά πουλιά που δεν είναι έτοιμα για αναπαραγωγή διασκορπίζονται σε όλη την Μεσόγειο. Ενώ πιστευόταν ότι την μετανάστευση την έκανε μέσω της Διώρυγας του Σουέζ πάνω από την Ερυθρά θάλασσα, σύγχρονη μελέτη με την βοήθεια εντοπισμού με δορυφορική τηλεμετρία έδειξε όπως φαίνετε και στο Σχήμα 4.1 ότι περνά πάνω από την Σαχάρα [4]. Είναι πιστοί στον σύντροφο τους και τείνουν να επιστρέφουν συνήθως στην ίδια θέση φωλιάσματος.

Ο Μαυροπετρίτης πέτα και περιπολεί για την τροφή του από ακόμα και ένα χιλιόμετρο πάνω από το έδαφος ή την επιφάνεια της θάλασσας αντίστοιχα και ορμά γραπώνοντας

την λεία τους. Συνηθίζουν να κυνηγούν κατά την διάρκεια του λυκόφωτος. Είναι κυρίως κοινωνικά πουλιά και συνήθως κινούνται σε μικρά σμήνη και κατά την μετανάστευση κινούνται με άλλα είδη από γεράκια.[16]. Κτίζουν τις φωλιές τους κυρίως σε γκρεμούς κοντά στην θάλασσα, κυρίως σε μικρές ακατοίκητες νησίδες. Στις μικρές νησίδες προτιμούν να κατοικούν στο κέντρο τους ενώ στα μεγαλύτερες στην ακτογραμμή. Όταν μεταναστεύουν στην Μαδαγασκάρη προτιμούν να κτίζουν τις φωλιές τους σε ανοικτές δασώδεις περιοχές [3].



Σχήμα 4.1 Η μετανάστευση 2 εμπείρων και 2 νεαρών Μαυροπετρίτων από την Ελλάδα στην Μαδαγασκάρη[3]

Οι κίνδυνοι που παρουσιάζουν είναι συνήθως είναι οι άνθρωποι οι οποίοι κυνηγούν τα αυγά και τους νεοσσούς για τροφή (Global Raptor Information Network 2015). Η αύξηση του τουρισμού στην Ελλάδα και στην Μεσόγειο κατά τους καλοκαιρινούς μήνες επηρεάζει αρνητικά την αναπαραγωγική τους φάση. Παρόλο που παγκόσμια ο πληθυσμός αυτός δεν απειλείται, στην Ελλάδα (πάνω από τον 80% του πληθυσμού βρίσκεται στην Ελλάδα) θεωρείται σπάνιο και υπό προστασία αφού η διαβίωση του

εξαρτάται από το αν στα νησιά που κατοικούν υπάρχουν οι κατάλληλες συνθήκες για να γίνει η αναπαραγωγή του.

Αυγά

Τα αυγά γεννιούνται στα μέσα του Ιουλίου [3]. Ο Μαυροπετρίτης κάνει από 0-3 αυγά με την μεγαλύτερη πιθανότητα να κάνει 3. Κατά ακρίβεια ο μέσος όρος αυγών των φωλιών είναι 2.38 ± 0.78 [12]. Ένας ανησυχητικός παράγοντας που παρατηρείτε τα τελευταία χρόνια είναι ότι το τσόφλι του αυγού έγινε πιο λεπτό αφού επηρεάζεται από την μόλυνση του περιβάλλοντος, έτσι καταστρέφονται πιο εύκολα από τον ήλιο [13].

Νεοσσοί

Οι νεοσσοί μεγαλώνουν από τα τέλη Αύγουστου μέχρι τέλη Σεπτεμβρίου καθώς υπάρχει έντονη μετανάστευση άλλων πουλιών την περίοδο αυτή και είναι η βασική τροφή των νεοσσών [3]. Ξεκινούν την μετανάστευση πριν αναπτύξουν πλήρως τα φτερά τους.

4.2 Ανάλυση των δεδομένων

Οι τρεις βασικές φάσεις στην ζωή του Μαυροπετρίτη είναι το Αυγό, ο Νεοσσός και ο Ενήλικας. Τα στατιστικά δεδομένα που ακολουθούν είναι με πιθανότητες πως μας τα πήραμε από το τμήμα Βιολογικών Επιστημών του Πανεπιστημίου Κύπρου, συγκεκριμένα με τον Δρ. Alexander N. G. Kirschel και τον διδακτορικό φοιτητή Θωμά Χατζηκυριάκου.

Το Αυγό, που είναι η πρώτη φάση του κύκλου ζωής του Μαυροπετρίτη, με την δημιουργία του έχει να αντιμετωπίσει του εξής κινδύνους και παρουσιάζονται παρακάτω μαζί με την πιθανότητα να συμβεί το κάθε ένα:

- Καταστροφή του αυγού από τρωκτικά 3%
- Το αυγό να μην γονιμοποιήθηκε 10%
- Να καταστραφεί από την έκθεση στον ήλιο 10%
- Καταστροφή από ανθρώπους 3%

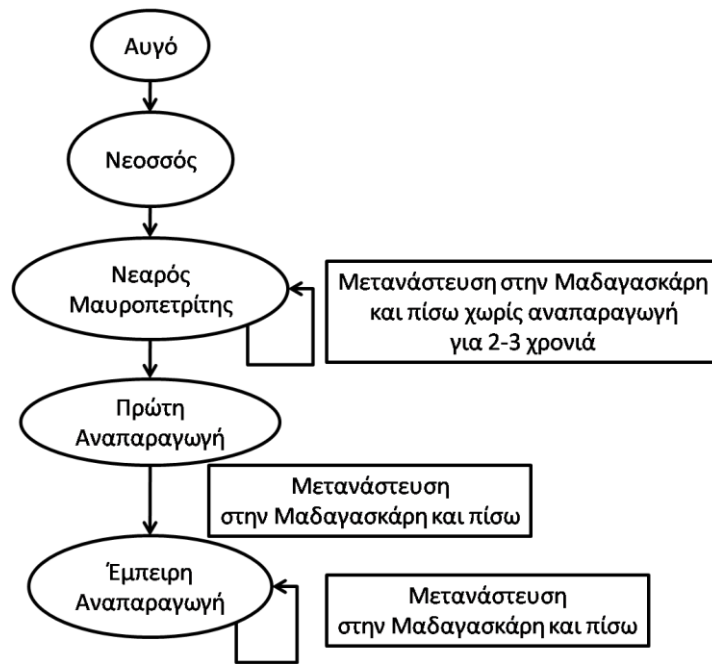
Αν το αυγό αυτό καταφέρει και επιβιώσει τους κινδύνους αυτούς με την υπόλοιπη πιθανότητα 74%, εκκολάπτετε και προχωρεί στην φάση του Νεοσσού.

Η φάση του νεοσσού αντιμετωπίζει με την σειρά της άλλους κινδύνους θνησιμότητας οι οποίοι είναι οι παρακάτω μαζί με την πιθανότητα τους να συμβούν:

- Κανιβαλισμός 5%
- Καταστροφή από ανθρώπους 3%
- Αρρώστιες 3%
- Τροφή για αρπακτικά 3%

Καταφέροντας να επιβιώσει από τους κινδύνους αυτούς ο νεοσσός προχώρα στην φάση του Ενήλικα με πιθανότητα 86%. Μπορούμε να παρατηρήσουμε ότι η επιβίωση και στις 2 φάσεις έχει μεγάλο ποσοστό, μεγαλύτερο από $\frac{3}{4}$, πράγμα ενθαρρυντικό για το είδος. Τα δεδομένα που έχουμε για την επόμενη φάση αφορούν ζευγάρια ενώ τα προηγούμενα αφορούσαν άτομα, θεωρήθηκε σαν σωστό αντίμετρο αυτής της μοντελοποίησης να συνεχίζουν από την φάση Νεοσσός στην φάση Ενήλικας μόνο περίπου το 70%, σωστή αναλογία θηλυκών αρσενικών θεωρώντας ότι κάθε Ενήλικας είναι μόνο τα θηλυκά.

Η φάση Ενήλικας χωρίζετε σε 2 υποφάσεις. Πριν την πρώτη υπόφαση ο Ενήλικας πρέπει να γίνει 2 ή 3 χρονών για να μπορεί να κάνει αναπαραγωγή αντιμετωπίζει αρκετούς κινδύνους και έχει πιθανότητα επιβίωσης μόλις 22% και το υπόλοιπο 78% πεθαίνει. Στην συνέχεια προχωρεί στην φάση τις πρώτης αναπαραγωγής και κάνει απογόνους με πιθανότητες, 40% να μην κάνει κανένα αυγό, με 9% να κάνει 1, με 16% να κάνει 2 και 35% να κάνει 3 απογόνους. Μετά το πέρας της υπόφασης της πρώτης αναπαραγωγής μπαίνει σε ένα πιο μικρό κύκλο ζωής, της έμπειρης αναπαραγωγής. Ξεκινώντας τον κύκλο αυτό κάθε ενήλικας αντιμετωπίζει τον θάνατο με πιθανότητα 13% ενώ το υπόλοιπο 87% προχώρα στην αναπαραγωγή. Τα ποσοστά αναπαραγωγής είναι πιο ενθαρρυντικά αφού για να μην κάνει αυγό είναι μόνο 30%, ενώ για ένα αυγό 10%, για δυο αυγά 18% και με την μεγαλύτερη πιθανότητα 42% να κάνει 3. Κάθε νέος απόγονος που δημιουργείτε είναι στην φάση του Αυγού. Τα δεδομένα ρύθμισης που έχουμε είναι πόσες φωλιές υπάρχουν κάθε χρονιά και πόσοι νεοσσοί κατάφεραν να πετάξουν την ίδια χρονιά. Μια γενική περιγραφή του κύκλου ζωής είναι η όπως το Σχήμα 4.2. Επίσης από κάθε κατάσταση έχει την πιθανότητα να πεθάνει.



Σχήμα 4.2 Ο Κύκλος της ζωής του Μαυροπετρίτη

Κεφάλαιο 5

Δημιουργία και Ανάλυση Μοντέλου

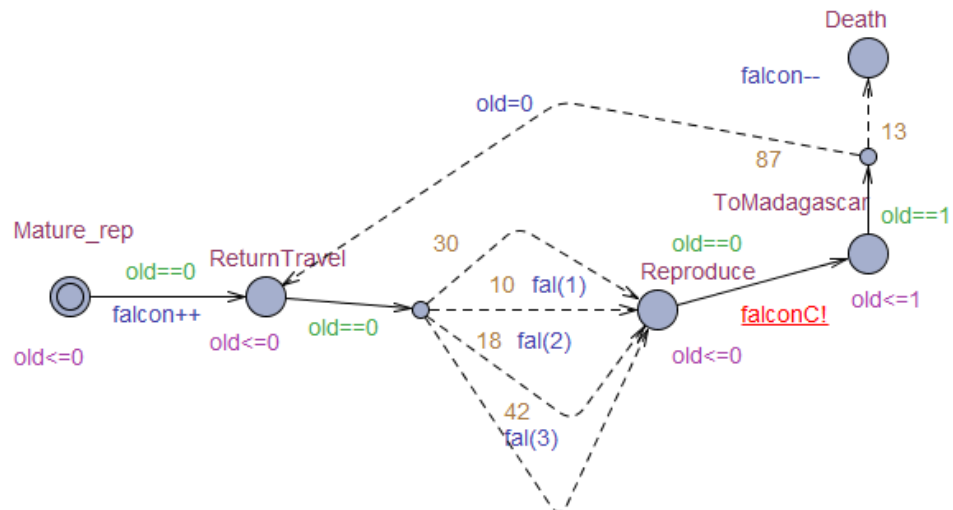
5.1 Δημιουργία μοντέλου	37
5.2 Δημιουργία Μοντέλου με Μέγιστο Αριθμό πληθυσμού	37
5.3 Δημιουργία Μοντέλου με Δυναμική Δημιουργία στοιχείων πληθυσμού	42
5.4 Ρύθμιση πιθανοτήτων για ανταπόκριση στην πραγματικότητα	45
5.5 Αποτελέσματα	47

5.1 Δημιουργία μοντέλου

Όπως φάνηκε από τα αποτελέσματα στο κεφάλαιο 3 το πιο κατάλληλο πρόγραμμα για την μοντελοποίηση και μοντελοέλεχο του συστήματος του κύκλου ζωής του Μαυροπετρίτη είναι το UPPAAL SMC. Έγιναν 2 προσεγγίσεις, βασισμένες στο κύκλο ζωής που αναλύθηκε στο 4.2, οι οποίες θα εξηγηθούν στην συνέχεια και οι δυο περιγραφικά στα υποκεφάλαια 5.2 και 5.3 αντίστοιχα. Η μια προσέγγιση είναι η δημιουργία μοντέλου με μέγιστο όριο στον πληθυσμό και η άλλη είναι δημιουργία στοιχείων του πληθυσμού με δυναμικό τρόπο. Και οι 2 προσεγγίσεις είναι αφαιρετικά μοντέλα με Εξήγηση Συμπεριφοράς αφού το UPPAAL SMC μας παρέχει την δυνατότητα αυτής της μοντελοποίησης, με Συνεχής Χώρο- Συνεχής Χρόνο. Στην πρώτη προσέγγιση ο Μαυροπετρίτης προσομοιώνεται σαν 2 διαφορετικά στοιχεία, τον αρχικό πληθυσμό και τον πληθυσμό που θα αναπαραχθεί μετά. Στην δεύτερη προσέγγιση κάθε κύρια φάση τις ζωής του Μαυροπετρίτη μοντελοποιείτε σαν ξεχωριστό στοιχείο μοντελοποίησης και όλα αλληλεπιδρούν μεταξύ τους με ένα στοιχείο Έλεγχου.

5.2 Δημιουργία Μοντέλου με Μέγιστο Αριθμό πληθυσμού

Αυτή η προσέγγιση απαιτεί να μπει ένα όριο στο μέγιστο αριθμό που μπορεί να πάρει ο πληθυσμός, και πιο συγκεκριμένα το μέγιστο αριθμό που μπορεί να είναι ζωντανά ταυτόχρονα ο κάθε οργανισμός στο σύστημα. Ο τρόπος που λειτουργεί το σύστημα αυτό κατά την αρχικοποίηση όλοι οι οργανισμοί και βρίσκονται σε μια φάση «αδράνειας» περιμένοντας να έρθει η σειρά τους να ενεργοποιηθούν. Υπάρχει ένα Template Starting Falcon που αντιπροσωπεύει τον αρχικό πληθυσμό ο οποίος ξεκινά στην φάση της έμπειρης αναπαραγωγής. Όπως βλέπουμε στο Σχήμα 5.1 ο αρχικός πληθυσμός για πρώτο βήμα που έχει να κάνει είναι να αναπαραχθεί και μετά να μπει στην φάση ToMadagascar. Εκεί περιμένει ένα ολόκληρο χρόνο και επιστρέφει στην κατάσταση ReturnTravel με την πιθανότητα να πεθάνει και επαναλαμβάνει τον κύκλο (βλέπε ΠΑΡΑΡΤΗΜΑ D).



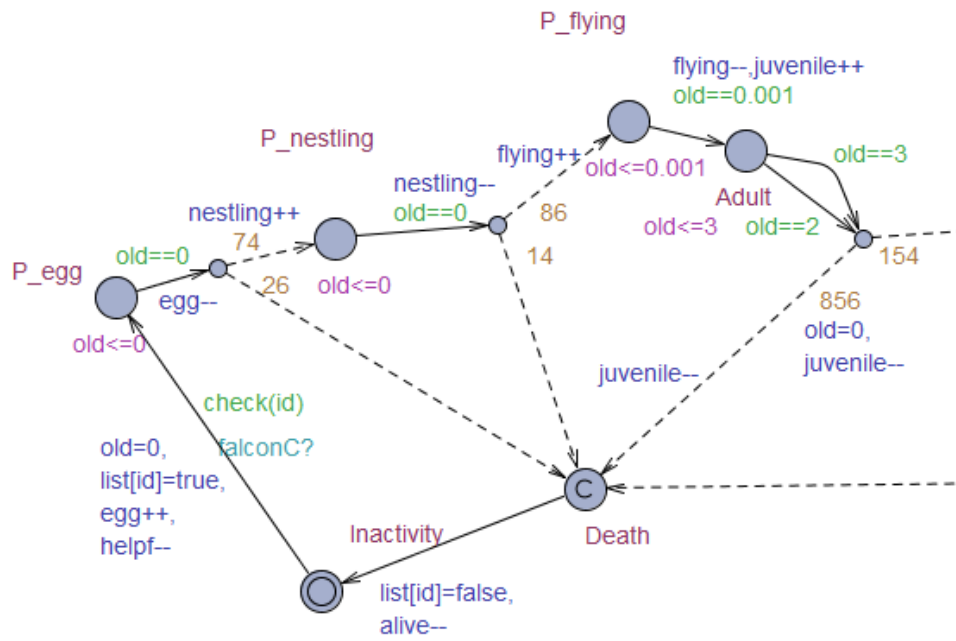
Σχήμα 5.1: Η μοντελοποίηση του Starting Falcon που μοντελοποιεί τον αρχικό πληθυσμό του συστήματος Μοντέλου με Μέγιστο Αριθμό.

Για να ξέρουν τα νέα Falcon να ενεργοποιηθούν στέλνει ένα broadcast μήνυμα στο κανάλι επικοινωνίας, και οποία είναι έτοιμα προχωρούν από την αδράνεια στην επομένη φάση. Υπάρχει ένας Boolean πίνακας `list[max]` που αντιπροσωπεύει ποια από τα Falcon είναι ζωντανά και ποια όχι, ανάλογα με την ταυτότητά τους. Η συνάρτηση `fal(int)` Σχήμα 5.2 μοντελοποιεί την αναπαραγωγή, αυξάνει δηλαδή αντίστοιχα τον αριθμό των αυγών για να ξέρουν πόσα νέα αυγά θα ενεργοποιηθούν.

```
void fal(int j){
```

}

Σχήμα 5.2: Η υλοποίηση της συνάρτησης `fal(int)`



Σχήμα 5.3: Η μοντελοποίηση στις 2 πρώτες φάσεις, του Αυγού και του Νεοσσού του τα Falcon

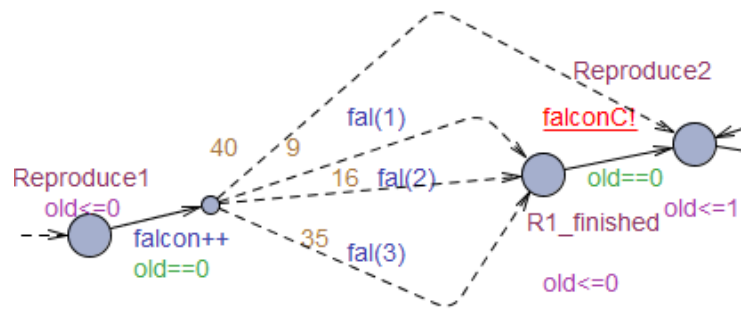
Στο Σχήμα 5.3 βλέπουμε τι πρώτες φάσεις της ζωής του Μαυροπετρίτη. Σε όλα τα Falcon Template ανατίθεται ένας αύξων αριθμός που αντιπροσωπεύει την ταυτότητα τους και βρίσκονται στην φάση Inactivity μόλις ξεκινήσει το σύστημα. Μόλις πάρουν όλα το broadcast μήνυμα για να ενεργοποιηθούν, τρέχουν την συνάρτηση `check(id)`

Σχήμα 5.4 η οποία ελέγχει τον αύξων αριθμό και απάντα αν θα προχωρήσει στην επόμενη φάση το Falcon ή όχι. Ελέγχει βασικά την μεταβλητή `helpf` και μετρά πόσα inactive υπάρχουν, από τον πίνακα `list[max]`, με μικρότερο `id` από το δικό του. Αν ο αριθμός των inactive είναι μικρότερα από το `helpf` τότε σημαίνει είναι έτοιμο να

προχωρήσει στην κατάσταση P_Egg. Όταν κάνει την μετάβαση ενημερώνει ότι είναι active και αυξάνει τον αριθμό των αυγών. Στην κατάσταση P_Egg δεν παραμένει χρόνο αφού τα δεδομένα που μας ενδιαφέρουν είναι τα πόσα θα πετάζουν στο τέλος της χρονιάς. Μετά μπαίνει στην διαδικασία να επιλέξει ή να πεθάνει από τους κινδύνους αντιμετωπίζει το αυγό ή να προχωρήσει στην επόμενη κατάσταση. Στην κατάσταση του νεοσσού (P_nestling) ,ούτε σε εκείνη περιμένει χρόνο, επιλεγεί με κάποια πιθανότητα αν θα επιβιώσει από τους κινδύνους που αντιμετωπίζουν οι νεοσσοί ή όχι. Αν επιβιώσει μπαίνει στην κατάσταση P_flying και περιμένει τον συμβολικό χρόνο 0.001 με σκοπό να μπορούν να μετρηθούν πόσα flying υπάρχουν, καθώς στις προηγούμενες φάσεις δεν περνά ο χρόνος πρέπει όλα να φτάσουν σε αυτή την κατάσταση, και μετά να μπει στη κατάσταση του Ενήλικα. Στην κατάσταση αυτή περιμένει 2 ή 3 χρόνια και είναι η φάση των μη-αναπαραγωγισμων Ενηλίκων. Τελειώνοντας την φάση αυτή προχωρά στην φάση της πρώτης αναπαραγωγής, ενώ πρώτα υποβληθεί στον κίνδυνο να πεθάνει κατά την διάρκεια των χρόνων αυτών. Συνδυάζετε με την πιθανότητα αν είναι θηλυκό έτσι ώστε να προσαρμοστούν τα στατιστικά δεδομένα που είχαμε πριν την φάση αυτή, ήταν για Falcon ενώ για τις επόμενες φάσεις είναι για φωλιές. Όταν πεθάνει σε κάθε κατάσταση προχωρά στην κατάσταση Death και απευθείας προχωρεί στην αδράνεια περιμένοντας να ενεργοποιηθεί ξανά ενημερώνοντας ότι δεν είναι ενεργό στοιχείο.

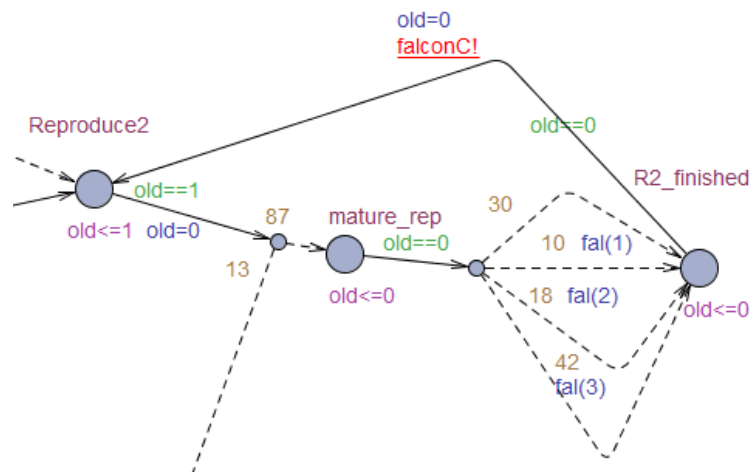
```
bool check(int id){
    bool d=false;int i=0;int temp=0;
    for(i=0;i<alive;i++){
        if(temp>=helpf){return false;}
        if(i==id){return true;}
        if(!list[i]){temp++;}
    }
    return d;
}
```

Σχήμα 5.4: Η υλοποίηση της συνάρτησης check(int)



Σχήμα 5.5: Η μοντελοποίηση της πρώτης αναπαραγωγής του Μαυροπετρίτη σε Μοντέλο με Μέγιστο Αριθμό

Κατά την διάρκεια αυτής της αναπαραγωγής, όπως φαίνεται στο Σχήμα 5.5, η οποία συμβαίνει μόνο μια φορά στον κύκλο της ζωής του Μαυροπετρίτη, δεν περιμένει κάποιο χρονικό διάστημα και προχωρά ανάλογα με τα αντίστοιχα στατιστικά στοιχεία στην αναπαραγωγή αυγών ή όχι. Αφού αυξάνει τον αριθμό των αυγών με την συνάρτηση $\text{fal}(\text{int})$ Σχήμα 5.2 καταλήγει στην κατάσταση Reproduce2 ενώ πρώτα έστειλε το μήνυμα να ενεργοποιηθούν ο αντίστοιχος αριθμός αυγών.



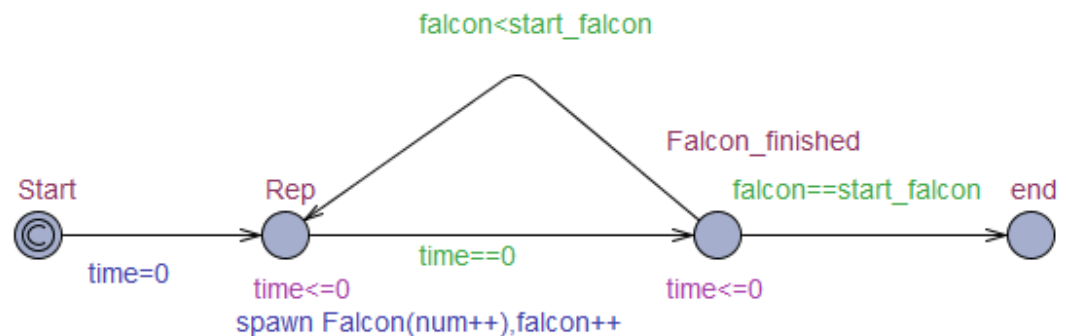
Σχήμα 5.6: Η μοντελοποίηση της έμπειρης αναπαραγωγής του Μαυροπετρίτη σε Μοντέλο με Μέγιστο Αριθμό

Μόλις ο Μαυροπετρίτης φτάσει στην κατάσταση Reproduce2 περιμένει μέχρι να περάσει ένας χρόνος (Σχήμα 5.6). Αυτό το διάστημα μοντελοποιείτε με την μετανάστευση στην Μαδαγασκάρη και την επιστροφή του πίσω. Αν επιβιώσει την

μετανάστευση, που αυτό μπορεί να συμβεί με αρκετά μεγάλη πιθανότητα, προχωρά στην έμπειρη αναπαραγωγή με αρκετά μεγαλύτερη πιθανότητα αναπαραγωγής των 3 αυγών. Όταν αυξάνει τον αριθμό των αυγών που πρέπει να γεννηθούν με την συνάρτηση `fal(int)` Σχήμα 5.2 στέλνει ένα μήνυμα προς όλα τα inactive Falcon για να ενεργοποιηθούν όσα πρέπει. Θέτει το χρονόμετρο του ξανά στο 0 για να περιμένει ένα χρόνο ξανά, για να μεταναστεύσει στην Μαδαγασκάρη και πίσω.

5.3 Δημιουργία Μοντέλου με Δυναμική Δημιουργία στοιχείων πληθυσμού

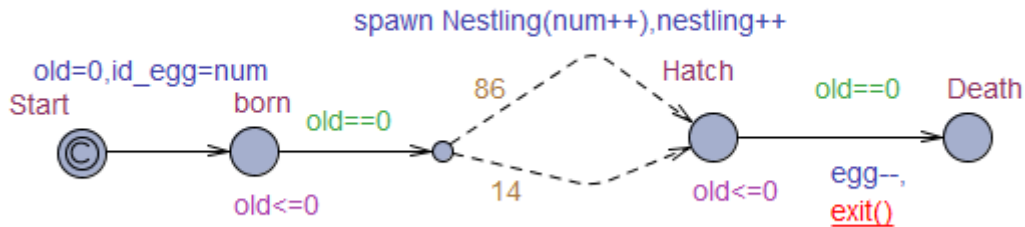
Η προσέγγιση αυτή είναι πιο ρεαλιστική μοντελοποίηση αφού η ιδιότητα του να γεννιέται και να πεθαίνει ένα στοιχείο υπάρχει έμφυτο στο σύστημα. Με της εντολές `spawn()` και `exit()` μπορούμε να αυξήσουμε τα στοιχεία του πληθυσμού δυναμικά, χωρίς να δεσμεύουμε χώρο στην μνήμη του υπολογιστή, φτάνοντας σε πολύ απομακρυσμένες χρονικά στιγμές πιο γρήγορα, μελετώντας ταυτόχρονα μεγαλύτερους αριθμούς πληθυσμού. Τα μοντέλα είναι μικρότερα και πιο άπλα περιγράφοντας μόνο τις πραγματικές διεργασίες του οργανισμού. Υπάρχει το Starting Template το οποίου και αυτού ο σκοπός του είναι να αρχικοποιήσει το σύστημα. Στο Σχήμα 5.7 φαίνεται η ποριά που έχει το Starting Template (βλέπε ΠΑΡΑΡΤΗΜΑ Ε).



Σχήμα 5.7: Η μοντελοποίηση του Starting Template σε Μοντέλο με δυναμική αύξηση των στοιχείων

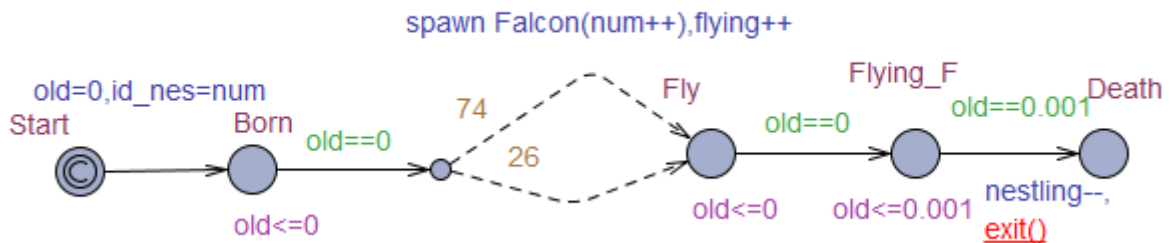
Δεν υπάρχουν εδώ κανάλια επικοινωνίας καθώς το Starting Template δημιουργεί κυριολεκτικά τα στοιχεία. Κάνει την επανάληψη όσο η μεταβλητή `falcon` είναι μικρότερη από την `start_falcon` και κάθε φορά δημιουργεί καινούργιο Falcon. Οι συνθήκες στις καταστάσεις και μεταβάσεις το αναγκάζουν να μην προχωρά χρόνος

μέχρι να δημιουργηθεί ο αρχικός πληθυσμός. Επίσης όλα τα στοιχεία αρχικοποιούνται με ένα κοινό αύξων αριθμό num.



Σχήμα 5.8: Η μοντελοποίηση του Αυγού σε Μοντέλο με δυναμική αύξηση των στοιχείων

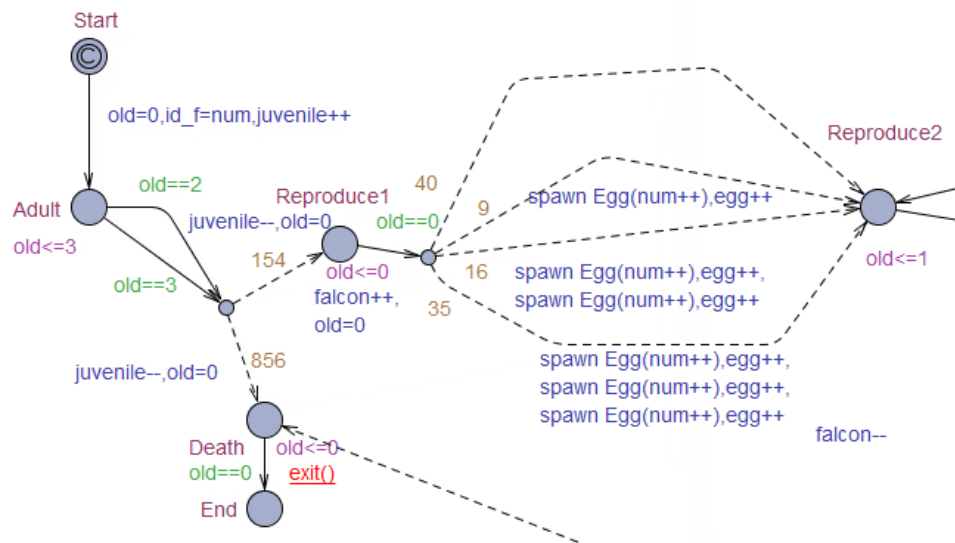
Το μοντέλο του Αυγού στο Σχήμα 5.8 δείχνει ακριβώς τις φάσεις ζωής του. Το στοιχείο αυτό μόλις δημιουργηθεί χωρίς να περάσει χρόνος εφαρμόζετε σε αυτό ο έλεγχος αν πέθανε από τους κινδύνους που αντιμετωπίζει το αυγό. Αν έζησε κάνει την δημιουργία του επόμενου στοιχείου Νεοσσού με την εκτέλεση του spawn() Nestling. Οποία από τις 2 επιλογές και να κάνει πάντα στο τέλος καταστρέφεται με το exit() αφού και στις 2 δεν υφίσταται πλέον το αυγό.



Σχήμα 5.9: Η μοντελοποίηση του Νεοσσού σε Μοντέλο με δυναμική αύξηση των στοιχείων

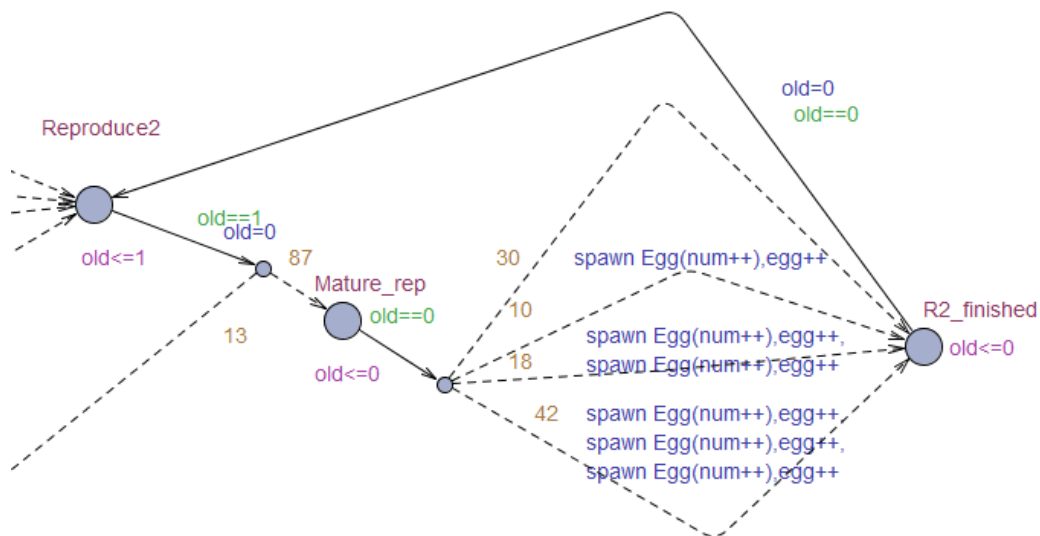
Το μοντέλο του Νεοσσού στο Σχήμα 5.9 δείχνει και αυτό τις πραγματικές φάσεις της ζωής του. Μόλις γεννηθεί επιλεγεί με κάποια πιθανότητα αν θα πεθάνει ή να προχωρήσει στην δημιουργία του επόμενου στοιχείου Ενήλικα, που συμβαίνει με την εκτέλεση του spawn() Falcon. Για να μετρήσουμε τους νεοσσούς που κατάφεραν να πετάξουν δημιουργούμε την κατάσταση Flying_F έτσι ώστε να περιμένουν όσα έφτασαν σε εκείνη την κατάσταση όλα τα υπόλοιπα τα οποία γεννηθήκαν την ίδια χρονιά και μπορεί να βρίσκονται σε προηγούμενη κατάσταση. Αυτό συμβαίνει αφού

στις προηγούμενες ο χρόνος δεν προχωρά .Στο τέλος το στοιχείο καταστρέφεται λόγω του ότι ο πραγματικός οργανισμός Νεοσσός δεν υφίσταται πλέον, είναι είτε Ενήλικας είτε νεκρός.



Σχήμα 5.10: Η μοντελοποίηση της πρώτης αναπαραγωγής Ενήλικων σε Μοντέλο με δυναμική αύξηση των στοιχείων

Η υπόφαση της πρώτης αναπαραγωγής ξεκινά αφού το Ενήλικο Falcon γίνει πρώτα 2 ή 3 χρόνων. Πριν προχωρήσει στην πρώτη αναπαραγωγή πρώτα πρέπει να υποβληθεί στον κίνδυνο να πεθάνει κατά την διάρκεια των χρόνων αυτών. Συνδυάζετε με την πιθανότητα αν είναι θηλυκό έτσι ώστε να προσαρμοστούν τα στατιστικά δεδομένα που είχαμε πριν την φάση αυτή, που ήταν για Falcon ενώ για τις επόμενες είναι για φωλιές. Αν ζήσει επιλεγεί πόσους απογόνους θα κάνει με τις ανάλογες πιθανότητες εκτελώντας την `spawn Egg(int)` και όπως φαίνετε και στο Σχήμα 5.10 και μπαίνει στην υπόφαση της έμπειρης αναπαραγωγής.



Σχήμα 5.11: Η μοντελοποίηση της έμπειρης αναπαραγωγής Ενηλίκων σε Μοντέλο με δυναμική αύξηση των στοιχείων

Η υπόφαση τις έμπειρης αναπαραγωγής, την οποία δείχνει και το Σχήμα 5.11 περιμένει μέχρι να περάσει ένας χρόνος. Αυτό το διάστημα μοντελοποιείτε με την μετανάστευση στην Μαδαγασκάρη και την επιστροφή του πίσω. Αν επιβιώσει την μετανάστευση, που αυτό μπορεί να συμβεί με αρκετά μεγάλη πιθανότητα, προχωρά στην έμπειρη αναπαραγωγή με αρκετά μεγαλύτερη πιθανότητα αναπαραγωγής με την μεγαλύτερη πιθανότητα των 3 αυγών.

Σε όλα τα σχήματα (Σχήμα 5.7 - Σχήμα 5.11) της δυναμικής προσέγγισης βλέπουμε ότι τα μοντέλα είναι πιο άπλα χωρίς συναρτήσεις για έλεγχο εγκυρότητας. Γεννιούνται και πεθαίνουν τα στοιχεία όπως ακριβώς οι πραγματικοί οργανισμοί και παράγουν ή μεταφέρονται στην επόμενη κατάσταση καταστρέφοντας την προηγούμενη, όπως συμβαίνει και στην υποθετικά στην πραγματικότητα.

5.4 Ρύθμιση πιθανοτήτων για ανταπόκριση στην πραγματικότητα

Τα στατιστικά δεδομένα που είχαμε στην διάθεση μας από το Τμήμα Βιολογίας, για ρύθμιση των πιθανοτήτων, ήταν για τις χρονιές 2012-2016 για τον πληθυσμό που αναπαράγεται στο Ακρωτήριο (Σχήμα 5.12) και στο Άσπρο μαζί με την Επισκοπή (Σχήμα 5.13). Στο Σχήμα 5.14 παρουσιάζονται τα αποτελέσματα για 4 χρονιές μοντελοποίησης, λαμβάνοντας βεβαία ότι για να μπορέσει να ρυθμιστεί σωστά η προσομοίωση έπρεπε

να ξεκινήσουμε να λαμβάνουμε υπόψη μας δεδομένα από τον 3 χρόνο, λόγο του ότι ο αρχικός πληθυσμός ήταν μόνο τα έμπειρα πούλια και δεν υπήρχαν αυτά που ήταν ηδύ 1-2 χρόνων και θα έκαναν την επομένη χρονιά την αναπαραγωγή. Το κάθε σχήμα παρουσιάζει για κάθε χρονιά πόσοι νεοσσοί πέταξαν, πόσες φωλιές υπήρχαν και την αναλογία τους. Θέσαμε σαν αρχικό πληθυσμό τον αριθμό 21, μιας και στο Σχήμα 5.12 αυτός είναι ο αρχικός πληθυσμός, και παρατηρήσαμε ότι οι νεοσσοί αντί να έχουν ένα ποσοστό αναλογίας 150%-200% που φαίνετε στα Σχήματα 5.12 και 5.13 για το 2012 είχε μόλις 117%. Επίσης η αύξηση των Νεοσσών που πέταξαν όσο και η αύξηση των φωλιών ήταν πολύ αργή.

	2012	2013	2014	2015	2016
Φωλιές	21	36	27	31	35
Νεοσσοί	43	54	50	53	61
Νεοσσοί / Φωλιές	200%	150%	185%	170%	174%

Σχήμα 5.12: Τα δεδομένα για τον πληθυσμό του Μαυροπετρίτη στο Ακρωτήρι, δείχνοντας τις φωλιές και πόσοι από τους νεοσσούς πέταξαν και η αναλογία Φωλιών ως προς Νεοσσούς

	2012	2013	2014	2015	2016
Φωλιές	32	50	47	39	38
Νεοσσοί	48	95	85	63	64
Νεοσσοί / Φωλιές	150%	190%	180%	161%	168%

Σχήμα 5.13: Τα δεδομένα για τον πληθυσμό του Μαυροπετρίτη στο Άσπρο και στη Επισκοπή, δείχνοντας τις φωλιές και πόσοι από τους νεοσσούς πέταξαν και η αναλογία Φωλιών ως προς Νεοσσούς

	4	5	6	7
Φωλιές	23	24	24	28
Νεοσσοί	27	28	29	31
Νεοσσοί / Φωλιές	117%	116%	120%	110%

Σχήμα 5.14: Τα αποτελέσματα της μοντελοποίησης του πληθυσμού του Μαυροπετρίτη και η αναλογία Φωλιών ως προς Νεοσσούς με τις κανονικές ποσοτικές τιμές

Μελετώντας καλύτερα τα στοιχεία που είχαμε για τον πληθυσμό αποφασίσαμε ότι οι αλλαγές που μπορούν να γίνουν είναι στη πιθανότητα πόσα αυγά μπορεί να κάνει κάθε φωλιά. Συγκεκριμένα αν αλλάζαμε τα δεδομένα της έμπειρης αναπαραγωγής σε 13% (30%) να μην κάνει αυγά, 9% (10%) να κάνει ένα αυγό, 25 % (18%) να κάνει δυο και 52 % (42%) να κάνει 3 αυγά, τα αποτελέσματα που παρουσιάζονται είναι πιο λογικά. Επίσης αν μειώναμε τους κινδύνους για καταστροφή από αυγά ως εξής:

- Καταστροφή του αυγού από τρωκτικά 2% (3%)
- Το αυγό να μην γονιμοποιήθηκε 5% (10%)
- Να καταστραφεί από την έκθεση στον ήλιο 5% (5%)
- Καταστροφή από ανθρώπους 2% (3%)

Και την φάση του νεοσσού εξής:

- Κανιβαλισμός 3% (5%)
- Καταστροφή από ανθρώπους 3% (στάσιμο)
- Αρρώστιες 2% (3%)
- Τροφή για αρπακτικά 2% (3%)

Αλλάζοντας αυτές τις πιθανότητες αλλάζουμε και τον ρυθμό των Ενηλίκων Μαυροπετρίτων έμμεσα αφού εξαρτούνται από τους Νεοσσούς. Οι αλλαγές αυτές έφεραν νέα αποτελέσματα αρκετά πιο αποδεκτά από τα προηγούμενα όπως παρουσιάζονται στο Σχήμα 5.15.

	4	5	6	7
Φωλιές	27	30	32	37
Νεοσσοί	44	48	50	62
Νεοσσοί / Φωλιές	162%	160%	156%	167%

Σχήμα 5.15: Τα αποτελέσματα της μοντελοποίησης του πληθυσμού του Μαυροπετρίτη και η αναλογία Φωλιών ως προς Νεοσσούς με τις αλλαγμένες ποσοτικές τιμές

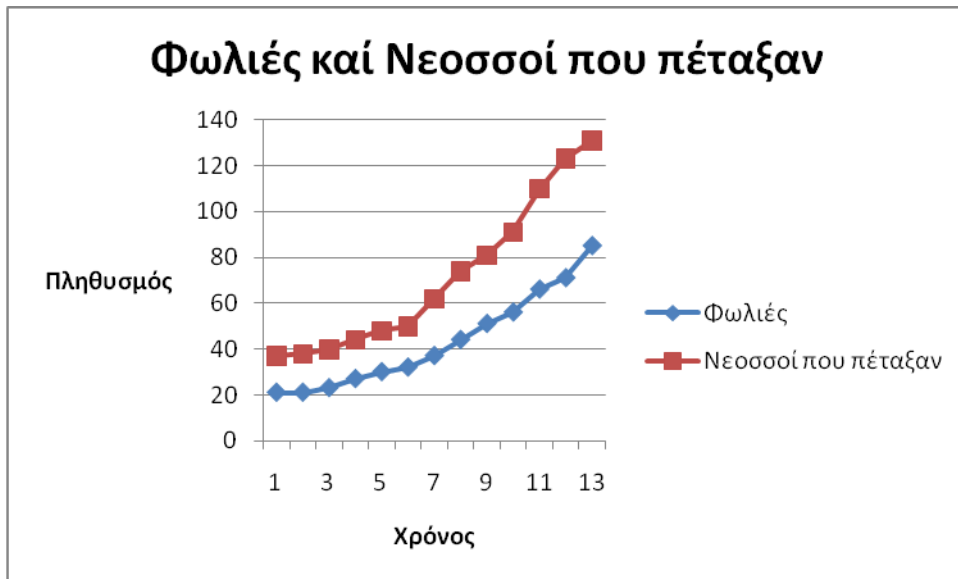
5.5 Αποτελέσματα

Παρατηρώντας και τα 2 μοντέλα παρουσιάζουν μια ίση μοντελοποίηση του πληθυσμού. Η διάφορα τους είναι ο χρόνος που θα τους πάρει να απαντήσουν στα Ερωτήματα που θέτουμε κάθε φορά. Μια γρήγορη εκτίμηση είναι ότι η Δυναμική Δημιουργία στοιχείων αφού δεν καταναλώνει χώρο στην μνήμη και μόνο όσα είναι δημιουργημένα την χρησιμοποιούν, θα είναι αρκετά πιο γρήγορη. Δυστυχώς όμως το πρόγραμμα UPPAAL δεν είναι πλήρως υλοποιημένο για μοντελοέλεγχο για την Δυναμική Δημιουργία Στοιχείων και γι αυτό δεν μπορέσαμε να τρέξουμε τον μοντελοελεκτή και για τα 2 περιπτώσεις. Καταφέραμε να τρέξουμε μια απλή προσημείωση χωρίς μοντελοέλεγχο για 100 χρονιά και για τα δυο και ανακαλύψαμε ότι η Μοντελοποίηση με αρχικό πληθυσμό ήταν πιο γρήγορη. Τα αποτελέσματα για σύγκριση που παρουσιάζονται στο Σχήμα 5.14 και στο Σχήμα 5.15 είναι με αυτή την προσέγγιση Μέγιστου Αριθμού πληθυσμού καθώς και τα δεδομένα που παρουσιάζουμε στην συνέχεια.

Τρέξαμε τα παρακάτω Ερωτήματα από 1 μέχρι 13 χρόνια. Το k αντιπροσωπεύει τον χρόνο κάθε φορά:

- $E[<=k,100](\text{max:} \text{flying})$
- $E[<=k,100](\text{max:} \text{falcon})$
- $E[<=k,100](\text{max:} \text{deadadults})$
- $E[<=k,100](\text{max:} \text{deadflying})$
- $E[<=k,100](\text{max:} \text{falcon} + \text{juveniles})$

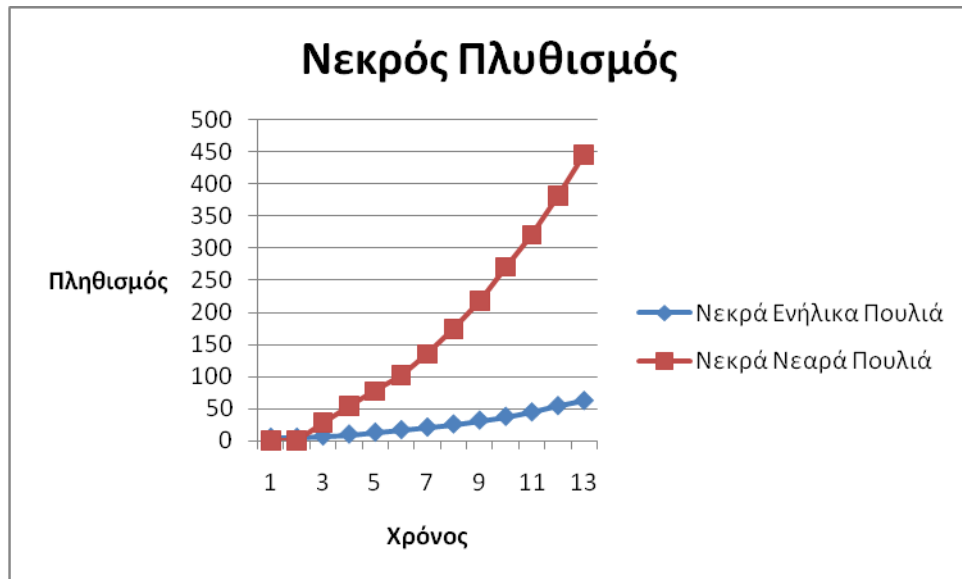
Από τα δεδομένα που πήραμε δημιουργήσαμε τις παρακάτω γραφικές παραστάσεις (παρουσιάζονται στο ΠΑΡΑΡΤΗΜΑ F):



Σχήμα 5.16: Η αύξηση του πληθυσμού των φωλιών και των πουλιών που πέταξαν κάθε χρονιά



Σχήμα 5.17: Ο συνολικός πληθυσμός των Ενηλίκων και Νεαρών πουλιών κάθε χρονιά



Σχήμα 5.18: Ο αριθμός των νεκρών κάθε χρόνο Ενηλίκων και Νεαρών Πουλιών

Κεφάλαιο 6

Συμπεράσματα

6.1 Αξιολόγηση Αποτελεσμάτων	51
6.2 Μελλοντικές Εργασίες	52

6.1 Αξιολόγηση Αποτελεσμάτων

Από τα αποτελέσματα που πήραμε για τον Κυπριακό πληθυσμό του Μαυροπετρίτη μπορούμε να πούμε ότι ο πληθυσμός θα συνεχίσει να μεγαλώνει με γοργούς ρυθμούς. Αρκετά ενθαρρυντικά τα αποτελέσματα της μοντελοποίησης γιατί δείχνουν ότι τα επόμενα 13 χρόνια ο πληθυσμός θα γίνει τέσσερις φορές του σημερινού πληθυσμού. Μάθαμε επίσης ότι οι κίνδυνοι που αντιμετωπίζουν τόσο τα Αυγά όσο και οι Νεοσσοί είναι σε μικρότερο βαθμό από όσο πίστευαν οι ερευνητές. Δεν θα αλλάξουν δραματικά τα αποτελέσματα αν μειώσουμε τον κίνδυνο καταστροφής είτε των Αυγών είτε των Νεοσσών. Θα ήταν κάλο αν μειώναμε την πιθανότητα θανάτου τα 2-3 χρόνια που δεν είναι ενεργό στην αναπαραγωγή μιας και μόνο 1 στα 4 να θα ζήσει, όμως δεν είναι δυνατό από το άνθρωπο αυτό μιας και στην φάση αυτή σκορπίζετε στην Μεσόγειο.

Από την Μελέτη για την βελτίωση της μεθοδολογίας για την Δημιουργία μοντέλου καταλήξαμε ότι η μοντελοποίηση μέσω του εργαλείου UPPAAL είναι καλύτερη μιας και παρέχει μια οπτική μοντελοποίηση σε ανάθεση με το PRISM. Είναι αρκετά πιο εύκολο στην κατανόηση της γλώσσας καθώς είναι σε μια μορφή απλοποιημένης C και τα ρολόγια του συστήματος είναι υλοποιημένα, χωρίς να πρέπει να τα δημιουργήσεις εσύ. Αυτοί οι τρεις παράγοντες οδηγούν στη αρκετά πιο εύκολη και απλή δημιουργία των μοντέλων, αφού μπορείς να μοντελοποιήσεις ακριβώς ένα αφαιρετικό σύστημα του κύκλου ζωής ενός πληθυσμού, να εκφράσεις τις μεταβάσεις από την μια κατάσταση στην άλλη όπως ακριβώς συμβαίνουν και τέλος μπορείς να αποσφαλματώσεις το

μοντέλο σου πολύ πιο εύκολα, με την βοήθεια του οπτικού εργαλείου Simulator, και αποδοτικά καθώς και να το εξηγήσεις σε ένα τρίτο άτομο πιο εύκολα.

6.2 Μελλοντικές Εργασίες

Η διπλωματική αυτή θα μπορούσε να επεκταθεί, όταν μπορέσουμε να πάρουμε τα αποτελέσματα από την προσέγγιση δημιουργίας δυναμικών στοιχείων, αφού υλοποιηθεί το εργαλείο από την UPPAAL πλήρως. Θα ήταν αρκετά ενδιαφέρον να μάθουμε αν τα 2 μοντέλα είναι πραγματικά ίσα μετά τον μοντελοέλεγχο και ποια από τις 2 προσεγγίσεις είναι πραγματικά η πιο γρήγορη.

Αρκετά ενδιαφέρον επίσης να μελετηθεί, αν τα αποτελέσματα έχουν μεγάλη απόκλιση από την πραγματικότητα και ποιος παράγοντας συνέτισε στην μεγάλη αυτή διαφορά. Θα ήταν ενδιαφέρον βέβαια να μαζευτούν σωστά τα αποτελέσματα, μιας και τα αποτελέσματα αυτά δεν μαζευτήκαν για αυτό τον λόγο και ήταν ελλιπές μερικά σημαντικά στοιχεία, να μαζευτούν στοιχεία όπως πόσα αυγά υπήρχαν σε κάθε φωλιά, ποια είναι τα ατομικά στατιστικά αναπαραγωγής και όχι για φωλιές. Αν ίσως προσθέταμε περισσότερα στοιχεία στο σύστημα για να αλληλεπιδράσουν μαζί του συγκεκριμένου πουλιού. Τέτοια στοιχεία θα ήταν ενδιαφέρον να μοντελοποιήσουμε οργανισμούς που θεωρούνται κίνδυνοι για τον Μαυροπετρίτη, στοιχεία που θεωρούνται τροφή για τον Μαυροπετρίτη, οι τοποθεσίες που κατοικεί, καιρικές συνθήκες και είδος της φωλιάς.

Τέλος είναι η βάση για μοντελοποίηση και άλλων πουλιών που θεωρούνται σημαντικά με σκοπό να τα σώσουμε.

Βιβλιογραφία

- [1] Bennet, B.S. “*Simulation fundamentals*”. Prentice Hall.1995.Print
- [2] Berec L.”*Techniques of spatially explicit individual-based models: construction, simulation and mean-field analysis*” 150:55-81.2002
- [3] Dimalexis A, Xirouchakis S, Portolou D, et al. “*J Ornithol*“. (2008) 149: 23. doi:10.1007/s10336-007-0207-4
- [4] Gschweng M, Kalko K.V E., Querner U. ”*All across Africa: highly individual migration routes of Eleonora's falcon*” 2008.DOI: 10.1098/rspb.2008.0575
- [5] Kassara, C., Dimalexis, A., Fric, J. et al. “*J Ornithol*“. (2012) 153: 663. doi:10.1007/s10336-011-0784-0
- [6] Odum E, Barrett G. “*Fundamentals of Ecology*”.Sinauer.1993.Print
- [7] Papamichael P. 2015 “*Σύγκριση των αλγεβρικών διεργασιών PALPS και S-PALPS για την ανάλυση οικολογικών συστημάτων μεσών μελέτης του κύκλου της ζωής του Μαυροπετρίτη*”
- [8] Philippou A., Toro M. and Antonaki M. “*Simulation and Verification in a Process Calculus for Spatially-Explicit Ecological Models*”. Sci. Ann. Comp. Sci.,23.1:119-167,2013.
- [9] Philippou A., and Toro, M., “*Process ordering in a process calculus for spatially-explicit ecological models. In: Software Engineering and Formal Methods*”. pages 345-361, Springer, 2014.
- [10] Toro M., Philippou A., Kassara C. and Sfenthourakis S. “*Synchronous parallel composition in a process calculus for ecological models. From individuals to populations*”. In Theoretical Aspects of Computing-ICTAC 2014, pages 424-441.Springer 2014
- [11] Roberts Nancy ,Andersen David “*Introduction to Computer Simulation*”.Addison-Wesley.1983.Print
- [12] Salah T, Menouar S. et al “*Breeding biology of Eleonoras Falcon, Falco eleonora Gene 1839(Accipitriformes Falcondae), in Northeast Algeria at Serigina Island*”. Biodiversity Journal.2013.4 (1):117-123
- [13] Walter H.”*Eleonora’s Falcon: Adaptations to prey and Habitat in a Social Raptor*”.1979. University of Chicago Press.

- [14] Weisberg, M. “*Forty years of ‘the strategy: Levins on model building and idealization’*”. Biol. Phil. 2006.21 623–645
- [15] Online documentation “*A Tutorial on Uppaal*”.
<http://people.cs.aau.dk/~adavid/publications/21-tutorial.pdf>
- [16] <http://www.iucnredlist.org/details/22696442/0>
- [17] Online PRISM documentation. <http://www.prismmodelchecker.org/doc/>
- [18] <http://www.uppaal.org/>

Παράρτημα Α

Κώδικας σε γλώσσά PRISM για την μοντελοποίηση του Parasitic Varroa

```
dtmc
const int N=2;
const int start0=0;
const int start1=N;

const int start2=-1;
const int start3=-1;
const int start4=-1;
const int start5=-1;
const int start6=-1;

const int T=100;
const double repr=0.3;

global id0:int init 0;
global id1:int init 1;
global id2:int init 2;
global id3:int init 3;
global id4:int init 4;
global id5:int init 5;
global id6:int init 6;
global id7:int init 7;
global id8:int init 8;
global id9:int init 9;

global Xparents:int init 123;
global Yparents:int init 123;

module master
num_mites:[0..T] init 2;
reprod:int init 0;
able_to_move:int init 1;
once:int init 1;

[] x0=x1 & y0=y1 & able_to_move=1 & moved0=1 -
>(num_mites'=num_mites+1)&(reprod'=1)&(able_to_move'=0)&(Xparents'=x0)&(Yparen
ts'=y0);
[] (x0!=x1 | y0!=y1)& able_to_move=1 & moved0=1->(able_to_move'=0);
```

```

[] able_to_move=0 & moved0=0->(able_to_move'=1);

endmodule

module s0
  x0:int init start0;
  y0:int init start0;
  born0:int init start0;
  moved0:int init 0;

  [born] born0!=-1 & reprod=1-> true;
  [born] born0= -1 & reprod=1 & num_mites<=id0 ->true ;
  [born] born0= -1 & reprod=1 & num_mites>id0 ->
  (x0'=Xparents)&(y0'=Yparents)&(born0'=1)&(moved0'=1);

  [move_mite] able_to_move=0 & moved0=1 & num_mites>id0 ->(moved0'=0);
  [move_mite] able_to_move=0 & moved0=0 & num_mites<=id0 ->true;

  [tick]start0=-1->true;
  [tick]able_to_move=1 & moved0=0 & num_mites>id0 & x0!=0 & x0!=N & y0!=0 &
  y0!=N-> //kentro
      0.25:(x0'=x0+1)&(moved0'=1)
    +0.25:(y0'=y0+1)&(moved0'=1)
    +0.25:(y0'=y0-1)&(moved0'=1)
    +0.25:(x0'=x0-1)&(moved0'=1);

  [tick]able_to_move=1 & moved0=0 & num_mites>id0 & x0=0 & y0!=0 & y0!=N-> //proti
  sira
      0.33:(x0'=x0+1)&(moved0'=1)
    +0.33:(y0'=y0+1)&(moved0'=1)
    +0.34:(y0'=y0-1)&(moved0'=1);

  [tick]able_to_move=1 & moved0=0 & num_mites>id0 & x0=N & y0!=0 & y0!=N->
  //teleftea sira
      0.33:(x0'=x0-1)&(moved0'=1)
    +0.33:(y0'=y0+1)&(moved0'=1)
    +0.34:(y0'=y0-1)&(moved0'=1);

  [tick]able_to_move=1 & moved0=0 & num_mites>id0 & x0!=N & x0!=0 & y0=0->
  //proti grammi
      0.33:(x0'=x0-1)&(moved0'=1)

```

```

+0.33: (x0'=x0+1) & (moved0'=1)
+0.34: (y0'=y0+1) & (moved0'=1);

[tick]able_to_move=1 & moved0=0 & num_mites>id0 & x0!=N & x0!=0 & y0=N->
//teleftea grammi
    0.33: (x0'=x0-1) & (moved0'=1)
+0.33: (x0'=x0+1) & (moved0'=1)
+0.34: (y0'=y0-1) & (moved0'=1);

[tick]able_to_move=1 & moved0=0 & num_mites>id0 & x0=N & y0=N-> //telefte kouti
    0.50: (x0'=x0-1) & (moved0'=1)
+0.50: (y0'=y0-1) & (moved0'=1);

[tick]able_to_move=1 & moved0=0 & num_mites>id0 & x0=N & y0=0-> //
    0.50: (x0'=x0-1) & (moved0'=1)
+0.50: (y0'=y0+1) & (moved0'=1);

[tick]able_to_move=1 & moved0=0 & num_mites>id0 & x0=0 & y0=0-> //proto kouti
    0.50: (x0'=x0+1) & (moved0'=1)
+0.50: (y0'=y0+1) & (moved0'=1);

[tick] able_to_move=1 & moved0=0 & num_mites>id0 & x0=0 & y0=N-> //
    0.50: (x0'=x0+1) & (moved0'=1)
+0.50: (y0'=y0-1) & (moved0'=1);

Endmodule

module s1 = s0 [ x0=x1, start0=start1,born0=born1,moved0=moved1, y0=y1,
id0=id1] endmodule
module s2 = s0 [ x0=x2, start0=start2,born0=born2,moved0=moved2, y0=y2,
id0=id2] endmodule
module s3 = s0 [ x0=x3, start0=start3,born0=born3,moved0=moved3, y0=y3,
id0=id3] endmodule
module s4 = s0 [ x0=x4, start0=start4,born0=born4,moved0=moved4, y0=y4,
id0=id4] endmodule
module s5 = s0 [ x0=x5, start0=start5,born0=born5,moved0=moved5, y0=y5,
id0=id5] endmodule
module s6 = s0 [ x0=x6, start0=start6,born0=born6,moved0=moved6, y0=y6,
id0=id6] endmodule

```

Παράρτημα Β

Κώδικας σε xml σε UPPAAL SMC για την μοντελοποίηση του Parasitic Varroa με καθορισμό αρχικού πληθυσμού

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE nta PUBLIC "-//Uppaal Team//DTD Flat System 1.1//EN"
'http://www.it.uu.se/research/group/darts/uppaal/flat-1_2.dtd'>
<nta>
    <declaration>// Place global declarations here.
    int num=1;
    int alived=2;
    const int max=10;
    const int N=3;
    int reprod=3;//out of 10

    urgent broadcast chan start;
    int cx[max+1];
    int cy[max+1];
    bool alive[max];
    bool rep[max];
    int f;
    void pop(int id,int r){
        f=id;
        if(r<=reprod){
            if(num<max){
                num++;
                alived++; }}}
    void check(int id){
        int i=0;
        for(i=id+1;i<=alived-1;i++){
            if(cx[id]==cx[i]){
                if(cy[id]==cy[i]){
                    rep[id]=true;}}}}
    void move(int id,int opt){
        if(opt==1){
            if(cx[id]>0){//left
                cx[id]=cx[id]-1;
            }
        }else if (opt==2){//up
            if(cy[id]>0){
                cy[id]=cy[id]-1;
            }
        }else if (opt==3){//right
```



```

        if(cx[id]&lt;N-1){
            cx[id]=cx[id]+1;
        }
    }else if (opt==4){//down
        if(cy[id]&lt;N-1){
            cy[id]=cy[id]+1;
        }
    }
}
</declaration>
<template>
    <name>StartMite</name>
    <parameter>int id,int x,int y</parameter>
    <declaration>clock time;</declaration>
    <location id="id0" x="-1105" y="-374">
        <name x="-1147" y="-425">ready_to_rep</name>
        <label kind="invariant" x="-1088" y="-374">time&lt;=2</label>
    </location>
    <location id="id1" x="-892" y="-280">
        <name x="-866" y="-280">poped</name>
        <committed/>
    </location>
    <location id="id2" x="-994" y="-306">
        <name x="-1028" y="-314">mid</name>
        <committed/>
    </location>
    <location id="id3" x="-841" y="-365">
        <name x="-807" y="-382">ready_to_pop</name>
        <committed/>
    </location>
    <location id="id4" x="-1470" y="-289">
        <name x="-1487" y="-340">start_mite_start</name>
        <committed/>
    </location>
    <location id="id5" x="-1249" y="-297">
        <name x="-1300" y="-331">alive_state</name>
        <label kind="invariant" x="-1292" y="-357">time&lt;=1</label>
    </location>
    <location id="id6" x="-1139" y="-306">
        <name x="-1130" y="-289">moved</name>
        <committed/>
    </location>
    <init ref="id4"/>
    <transition>
        <source ref="id0"/>

```

```

        <target ref="id2"/>
        <label kind="guard" x="-1062" y="-408">time==2</label>
        <label kind="assignment" x="-1096" y="-
340">check(id)</label>
    </transition>
    <transition>
        <source ref="id6"/>
        <target ref="id0"/>
    </transition>
    <transition>
        <source ref="id1"/>
        <target ref="id2"/>
        <label kind="synchronisation" x="-943" y="-
331">start!</label>
        <label kind="assignment" x="-960" y="-
272">rep[id]=false</label>
    </transition>
    <transition>
        <source ref="id3"/>
        <target ref="id1"/>
        <label kind="select" x="-824" y="-
340">reprda:int[1,10]</label>
        <label kind="assignment" x="-858" y="-
323">pop(id,reprda)</label>
    </transition>
    <transition>
        <source ref="id2"/>
        <target ref="id3"/>
        <label kind="guard" x="-985" y="-454">rep[id]</label>
        <nail x="-1003" y="-425"/>
        <nail x="-875" y="-416"/>
    </transition>
    <transition>
        <source ref="id4"/>
        <target ref="id5"/>
        <label kind="guard" x="-1435" y="-327">id<num</label>
        <label kind="assignment" x="-1435" y="-
293">cx[id]=x,cy[id]=y,
alive[id]=true,time=0</label>
    </transition>
    <transition>
        <source ref="id2"/>
        <target ref="id5"/>
        <label kind="assignment" x="-1190" y="-119">time=0</label>
        <nail x="-1002" y="-152"/>

```

```

        <nail x="-1215" y="-135"/>
    </transition>
    <transition>
        <source ref="id5"/>
        <target ref="id6"/>
        <label kind="select" x="-1231" y="-
352">y4:int[1,4]</label>
        <label kind="guard" x="-1224" y="-331">time==1</label>
        <label kind="assignment" x="-1224" y="-
297">move(id,y4)</label>
    </transition>
</template>
<template>
    <name x="5" y="5">Mite</name>
    <parameter>int id</parameter>
    <declaration>clock time;
</declaration>
    <location id="id7" x="-1071" y="-365">
        <name x="-1113" y="-399">ready_to_rep</name>
        <label kind="invariant" x="-1081" y="-
348">time<=2</label>
    </location>
    <location id="id8" x="-884" y="-289">
        <name x="-858" y="-289">poped</name>
        <committed/>
    </location>
    <location id="id9" x="-841" y="-365">
        <name x="-824" y="-382">ready_to_pop</name>
        <committed/>
    </location>
    <location id="id10" x="-994" y="-306">
        <name x="-1037" y="-306">mid</name>
        <committed/>
    </location>
    <location id="id11" x="-1130" y="-306">
        <name x="-1113" y="-306">moved</name>
        <committed/>
    </location>
    <location id="id12" x="-1249" y="-297">
        <name x="-1300" y="-331">alive_state</name>
        <label kind="invariant" x="-1292" y="-
357">time<=1</label>
    </location>
    <location id="id13" x="-1470" y="-289">
        <name x="-1504" y="-323">mite_start</name>

```

```

                <label kind="invariant" x="-1504" y="-
272">num<math>id</math></label>
            </location>
            <init ref="id13"/>
            <transition>
                <source ref="id7"/>
                <target ref="id10"/>
                <label kind="guard" x="-1053" y="-369">time==2</label>
                <label kind="assignment" x="-1079" y="-
331">check(id)</label>
            </transition>
            <transition>
                <source ref="id11"/>
                <target ref="id7"/>
            </transition>
            <transition>
                <source ref="id8"/>
                <target ref="id10"/>
                <label kind="synchronisation" x="-943" y="-
331">start!</label>
                <label kind="assignment" x="-952" y="-
272">rep[id]=false</label>
            </transition>
            <transition>
                <source ref="id9"/>
                <target ref="id8"/>
                <label kind="select" x="-850" y="-
348">reprda:int[1,10]</label>
                <label kind="assignment" x="-858" y="-
323">pop(id,reprda)</label>
            </transition>
            <transition>
                <source ref="id10"/>
                <target ref="id9"/>
                <label kind="guard" x="-985" y="-454">rep[id]</label>
                <nail x="-1003" y="-425"/>
                <nail x="-875" y="-416"/>
            </transition>
            <transition>
                <source ref="id13"/>
                <target ref="id12"/>
                <label kind="guard" x="-1435" y="-327">id==num</label>
                <label kind="synchronisation" x="-1435" y="-
310">start?</label>

```

```

        <label kind="assignment" x="-1435" y="-
293">cx[id]=cx[f],cy[id]=cy[f],
alive[id]=true,time=0</label>
    </transition>
    <transition>
        <source ref="id10"/>
        <target ref="id12"/>
        <label kind="assignment" x="-1190" y="-119">time=0</label>
        <nail x="-1002" y="-152"/>
        <nail x="-1215" y="-135"/>
        <nail x="-1241" y="-246"/>
    </transition>
    <transition>
        <source ref="id12"/>
        <target ref="id11"/>
        <label kind="select" x="-1231" y="-
352">y4:int[1,4]</label>
        <label kind="guard" x="-1224" y="-331">time==1</label>
        <label kind="assignment" x="-1224" y="-
297">move(id,y4)</label>
    </transition>
</template>
<system>// Place template instantiations here.

```

```

P0 = StartMite(0,0,0);
P1 = StartMite(1,N-1,N-1);
P2 = Mite(2);
P3 = Mite(3);
P4 = Mite(4);
P5 = Mite(5);
P6 = Mite(6);
P7 = Mite(7);
// List one or more processes to be composed into a system.
system P0,P1,P2,P3,P4,P5,P6,P7;</system>
    <queries>
        <query>
            <formula>E<math>\leq 1</math>;deadlock
            </formula>
            <comment>
            </comment>
        </query>
        <query>
            <formula>simulate 1[<math>\leq 100</math>]{num}
            </formula>
            <comment>

```

```
        </comment>
      </query>
    </queries>
  </nta>
```

Παράρτημα C

Κώδικας σε xml σε UPPAAL SMC για την μοντελοποίηση του Parasitic Varroa με δυναμική αύξηση του πληθυσμού

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE nta PUBLIC "-//Uppaal Team//DTD Flat System 1.1//EN"
'http://www.it.uu.se/research/group/darts/uppaal/flat-1_2.dtd'>
<nta>
    <declaration>// Place global declarations here.
dynamic DynamMite(int id,int s);
const int N=16;
const int hal=4;
int table[N];
int num=0;

int move(int opt,int place){
    int next=place;
    if(opt==1){//left
        if(place%hal!=0){
            table[next]--;
            next--;
            table[next]++;
        }
    }else if(opt==2){//up
        if(place>=hal){
            table[next]--;
            next=next-hal;
            table[next]++;
        }
    }else if(opt==3){//right
        if(place%hal!=hal-1){
            table[next]--;
            next++;
            table[next]++;
        }
    }else if(opt==4){//down
        if(N-place>=hal){
            table[next]--;
            next=next+hal;
            table[next]++;
        }
    }
}
```

```

return next;
}
</declaration>
    <template>
        <name>DynamMite</name>
        <parameter>int id,int s</parameter>
        <declaration>clock time;
int place=0;</declaration>
        <location id="id0" x="-1887" y="-178">
            <name x="-1897" y="-212">born</name>
            <committed/>
        </location>
        <location id="id1" x="-1130" y="-59">
            <name x="-1096" y="-85">rest</name>
            <committed/>
        </location>
        <location id="id2" x="-1334" y="-25">
            <name x="-1385" y="-25">rep_7</name>
            <committed/>
        </location>
        <location id="id3" x="-1317" y="-119">
            <name x="-1351" y="-171">rep_3</name>
            <committed/>
        </location>
        <location id="id4" x="-1479" y="-136">
            <name x="-1530" y="-128">rep_y</name>
            <label kind="invariant" x="-1488" y="-
119">time<=2</label>
        </location>
        <location id="id5" x="-1241" y="-238">
            <name x="-1251" y="-272">rep_n</name>
            <label kind="invariant" x="-1258" y="-
298">time<=2</label>
        </location>
        <location id="id6" x="-1377" y="-238">
            <name x="-1387" y="-272">ch_for_rep</name>
            <committed/>
        </location>
        <location id="id7" x="-1624" y="-162">
            <name x="-1634" y="-196">m_4</name>
            <committed/>
        </location>
        <location id="id8" x="-1598" y="-230">
            <name x="-1608" y="-264">m_3</name>
            <committed/>

```



```

</location>
<location id="id9" x="-1581" y="-272">
    <name x="-1591" y="-306">m_2</name>
    <committed/>
</location>
<location id="id10" x="-1607" y="-340">
    <name x="-1617" y="-374">m_1</name>
    <committed/>
</location>
<location id="id11" x="-1793" y="-178">
    <name x="-1802" y="-229">ready_to_move</name>
    <label kind="invariant" x="-1861" y="-
229">time<=1</label>
</location>
<branchpoint id="id12" x="-1393" y="-93">
</branchpoint>
<branchpoint id="id13" x="-1708" y="-264">
</branchpoint>
<init ref="id0"/>
<transition>
    <source ref="id5"/>
    <target ref="id1"/>
    <label kind="guard" x="-1248" y="-216">time==2</label>
    <nail x="-1130" y="-170"/>
</transition>
<transition>
    <source ref="id0"/>
    <target ref="id11"/>
    <label kind="assignment" x="-1878" y="-
144">table[s]++,place=s,time=0</label>
</transition>
<transition>
    <source ref="id1"/>
    <target ref="id11"/>
    <label kind="assignment" x="-1444" y="38">time=0</label>
    <nail x="-1130" y="-42"/>
    <nail x="-1156" y="51"/>
    <nail x="-1462" y="25"/>
    <nail x="-1742" y="-34"/>
</transition>
<transition>
    <source ref="id2"/>
    <target ref="id1"/>
    <label kind="assignment" x="-1300" y="-34">spawn
DynamMite(num++,place),spawn DynamMite(num++,place)</label>

```

```

</transition>
<transition>
    <source ref="id3"/>
    <target ref="id1"/>
    <label kind="assignment" x="-1283" y="-127">spawn
DynamMite(num++,place)</label>
</transition>
<transition>
    <source ref="id12"/>
    <target ref="id2"/>
    <label kind="probability" x="-1363" y="-59">7</label>
</transition>
<transition>
    <source ref="id12"/>
    <target ref="id3"/>
    <label kind="probability" x="-1375" y="-105">3</label>
</transition>
<transition>
    <source ref="id4"/>
    <target ref="id12"/>
    <label kind="guard" x="-1428" y="-153">time==2</label>
</transition>
<transition>
    <source ref="id6"/>
    <target ref="id4"/>
    <label kind="guard" x="-1419" y="-
212">table[place]&gt;0</label>
</transition>
<transition>
    <source ref="id6"/>
    <target ref="id5"/>
</transition>
<transition>
    <source ref="id7"/>
    <target ref="id6"/>
    <label kind="assignment" x="-1555" y="-
188">place=move(4,place)</label>
</transition>
<transition>
    <source ref="id8"/>
    <target ref="id6"/>
    <label kind="assignment" x="-1589" y="-
230">place=move(3,place)</label>
</transition>
<transition>

```

```

        <source ref="id9"/>
        <target ref="id6"/>
        <label kind="assignment" x="-1564" y="-
280">place=move(2,place)</label>
    </transition>
    <transition>
        <source ref="id10"/>
        <target ref="id6"/>
        <label kind="assignment" x="-1538" y="-
323">place=move(1,place)</label>
    </transition>
    <transition>
        <source ref="id13"/>
        <target ref="id7"/>
        <label kind="probability" x="-1690" y="-213">1</label>
    </transition>
    <transition>
        <source ref="id13"/>
        <target ref="id8"/>
        <label kind="probability" x="-1690" y="-238">1</label>
    </transition>
    <transition>
        <source ref="id13"/>
        <target ref="id9"/>
        <label kind="probability" x="-1690" y="-268">1</label>
    </transition>
    <transition>
        <source ref="id13"/>
        <target ref="id10"/>
        <label kind="probability" x="-1690" y="-302">1</label>
    </transition>
    <transition>
        <source ref="id11"/>
        <target ref="id13"/>
        <label kind="guard" x="-1775" y="-212">time==1</label>
    </transition>
</template>
<template>
    <name x="5" y="5">Master</name>
    <declaration>// Place local declarations here.
</declaration>
    <location id="id14" x="-331" y="-42">
        <name x="-339" y="-85">end</name>
    </location>
    <location id="id15" x="-484" y="-42">

```

```

        <name x="-494" y="-76">start</name>
        <committed/>
    </location>
    <init ref="id15"/>
    <transition>
        <source ref="id15"/>
        <target ref="id14"/>
        <label kind="assignment" x="-595" y="-25">spawn
DynamMite(num++,0),spawn DynamMite(num++,N-1)</label>
    </transition>
</template>
<system>// Place template instantiations here.
Process = Master();
// List one or more processes to be composed into a system.
system Process;
</system>
    <queries>
        <query>
            <formula>simulate 1[&lt;=10]{num}
            </formula>
            <comment>
            </comment>
        </query>
    </queries>
</nta>

```

Παράρτημα D

Κώδικας σε xml σε UPPAAL SMC για την μοντελοποίηση της προσέγγισης με Μέγιστο Αρχικό Αριθμό

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE nta PUBLIC "-//Uppaal Team//DTD Flat System 1.1//EN"
'http://www.it.uu.se/research/group/darts/uppaal/flat-1_2.dtd'>
<nta>
    <declaration>// Place global declarations here.

    int falcon=0;
    int flying=0;
    int juvenile=0;
    int helpf=0;
    int nestling=0;
    int egg=0;

    int deadnestling=0;
    int deadfalcon=0;
    const int max=500;
    urgent broadcast chan falconC;

    int alive=0;

    bool list[max];

    bool check(int id){
        bool d=false;
        int i=0;
        int temp=0;
        for(i=0;i<alive;i++){
            if(temp>=helpf){
                return false;
            }
            if(i==id){
                return true;
            }
            if(!list[i]){
                temp++;
            }
        }
        return d;
    }
```

```

}

void fal(int j){

    int i=0;
    for(i=1;i<=j;i++){
        if(alive<max){
            helpf++;
            alive++;
        }
    }

}

</declaration>
    <template>
        <name x="5" y="5">Falcon</name>
        <parameter>int id</parameter>
        <declaration>// Place local declarations here.

clock old;

</declaration>
        <location id="id0" x="-1649" y="-221">
            <name x="-1700" y="-289">P_flying</name>
            <label kind="invariant" x="-1674" y="-
204">old<=0.001</label>
        </location>
        <location id="id1" x="-1419" y="-161">
            <name x="-1470" y="-203">Reproduce1</name>
            <label kind="invariant" x="-1444" y="-
186">old<=0</label>
        </location>
        <location id="id2" x="-1819" y="-153">
            <name x="-1862" y="-221">P_nestling</name>
            <label kind="invariant" x="-1829" y="-
136">old<=0</label>
        </location>
        <location id="id3" x="-1946" y="-119">
            <name x="-2006" y="-144">P_egg</name>
            <label kind="invariant" x="-1971" y="-
102">old<=0</label>
        </location>

```

```

        <location id="id4" x="-1827" y="51">
            <name x="-1810" y="8">Inactivity</name>
        </location>
        <location id="id5" x="-1173" y="-187">
            <name x="-1215" y="-170">R1_finished</name>
            <label kind="invariant" x="-1173" y="-
136">old<math>\leq 0</math></label>
        </location>
        <location id="id6" x="-663" y="-161">
            <name x="-705" y="-229">R2_finished</name>
            <label kind="invariant" x="-673" y="-
144">old<math>\leq 0</math></label>
        </location>
        <location id="id7" x="-892" y="-170">
            <name x="-902" y="-204">Mature_rep</name>
            <label kind="invariant" x="-902" y="-
153">old<math>\leq 0</math></label>
        </location>
        <location id="id8" x="-1054" y="-212">
            <name x="-1096" y="-263">Reproduce2</name>
            <label kind="invariant" x="-1071" y="-
195">old<math>\leq 1</math></label>
        </location>
        <location id="id9" x="-1581" y="-204">
            <name x="-1598" y="-187">Adult</name>
            <label kind="invariant" x="-1640" y="-
170">old<math>\leq 3</math></label>
        </location>
        <location id="id10" x="-1674" y="-8">
            <name x="-1674" y="9">Death</name>
            <committed/>
        </location>
        <branchpoint id="id11" x="-1725" y="-161">
        </branchpoint>
        <branchpoint id="id12" x="-1887" y="-136">
        </branchpoint>
        <branchpoint id="id13" x="-807" y="-161">
        </branchpoint>
        <branchpoint id="id14" x="-935" y="-178">
        </branchpoint>
        <branchpoint id="id15" x="-1351" y="-170">
        </branchpoint>
        <branchpoint id="id16" x="-1513" y="-153">
        </branchpoint>
        <init ref="id4"/>

```

```

<transition>
  <source ref="id0"/>
  <target ref="id9"/>
  <label kind="guard" x="-1623" y="-246">old==0.001</label>
  <label kind="assignment" x="-1623" y="-263">flying--
,juvenile++</label>
</transition>
<transition>
  <source ref="id11"/>
  <target ref="id0"/>
  <label kind="assignment" x="-1725" y="-
212">flying++</label>
  <label kind="probability" x="-1700" y="-178">90</label>
</transition>
<transition>
  <source ref="id16"/>
  <target ref="id1"/>
  <label kind="assignment" x="-1547" y="-110">old=0,
juvenile--</label>
  <label kind="probability" x="-1479" y="-153">22</label>
</transition>
<transition>
  <source ref="id1"/>
  <target ref="id15"/>
  <label kind="guard" x="-1394" y="-195">old==0</label>
  <label kind="assignment" x="-1360" y="-
144">falcon++</label>
</transition>
<transition>
  <source ref="id9"/>
  <target ref="id16"/>
  <label kind="guard" x="-1530" y="-212">old==3</label>
  <nail x="-1530" y="-195"/>
</transition>
<transition>
  <source ref="id11"/>
  <target ref="id10"/>
  <label kind="probability" x="-1708" y="-153">10</label>
</transition>
<transition>
  <source ref="id2"/>
  <target ref="id11"/>
  <label kind="guard" x="-1785" y="-178">old==0</label>
  <label kind="assignment" x="-1785" y="-195">nestling--
</label>

```



```

</transition>
<transition>
    <source ref="id12"/>
    <target ref="id10"/>
    <label kind="probability" x="-1861" y="-136">10</label>
</transition>
<transition>
    <source ref="id12"/>
    <target ref="id2"/>
    <label kind="assignment" x="-1904" y="-
178">nestling++</label>
    <label kind="probability" x="-1878" y="-161">90</label>
</transition>
<transition>
    <source ref="id3"/>
    <target ref="id12"/>
    <label kind="guard" x="-1946" y="-153">old==0</label>
    <label kind="assignment" x="-1921" y="-127">egg--</label>
</transition>
<transition>
    <source ref="id4"/>
    <target ref="id3"/>
    <label kind="guard" x="-1895" y="-68">check(id)</label>
    <label kind="synchronisation" x="-1887" y="-
42">falconC?</label>
    <label kind="assignment" x="-1955" y="-42">old=0,
list[id]=true,
egg++,
helpf--</label>
</transition>
<transition>
    <source ref="id14"/>
    <target ref="id10"/>
    <label kind="assignment" x="-1300" y="-25">falcon--
</label>
    <label kind="probability" x="-969" y="-170">13</label>
    <nail x="-994" y="-17"/>
</transition>
<transition>
    <source ref="id15"/>
    <target ref="id8"/>
    <label kind="probability" x="-1334" y="-255">35</label>
    <nail x="-1224" y="-280"/>
</transition>
<transition>

```

```

        <source ref="id5"/>
        <target ref="id8"/>
        <label kind="guard" x="-1173" y="-187">old==0</label>
        <label kind="synchronisation" x="-1190" y="-
238">falconC!</label>
    </transition>
    <transition>
        <source ref="id15"/>
        <target ref="id5"/>
        <label kind="assignment" x="-1274" y="-229">fal(1)</label>
        <label kind="probability" x="-1308" y="-212">11</label>
        <nail x="-1198" y="-221"/>
    </transition>
    <transition>
        <source ref="id15"/>
        <target ref="id5"/>
        <label kind="assignment" x="-1274" y="-153">fal(3)</label>
        <label kind="probability" x="-1300" y="-161">34</label>
        <nail x="-1215" y="-110"/>
    </transition>
    <transition>
        <source ref="id10"/>
        <target ref="id4"/>
        <label kind="assignment" x="-1793" y="42">list[id]=false,
alive--</label>
    </transition>
    <transition>
        <source ref="id15"/>
        <target ref="id5"/>
        <label kind="assignment" x="-1257" y="-195">fal(2)</label>
        <label kind="probability" x="-1283" y="-195">20</label>
    </transition>
    <transition>
        <source ref="id6"/>
        <target ref="id8"/>
        <label kind="guard" x="-772" y="-263">old==0</label>
        <label kind="synchronisation" x="-858" y="-
314">falconC!</label>
        <label kind="assignment" x="-858" y="-331">old=0</label>
        <nail x="-790" y="-297"/>
    </transition>
    <transition>
        <source ref="id13"/>
        <target ref="id6"/>
        <label kind="assignment" x="-748" y="-153">fal(2)</label>

```

```

        <label kind="probability" x="-773" y="-153">25</label>
        <nail x="-714" y="-93"/>
    </transition>
    <transition>
        <source ref="id13"/>
        <target ref="id6"/>
        <label kind="assignment" x="-756" y="-94">fal(3)</label>
        <label kind="probability" x="-756" y="-110">53</label>
        <nail x="-722" y="-34"/>
    </transition>
    <transition>
        <source ref="id13"/>
        <target ref="id6"/>
        <label kind="probability" x="-799" y="-212">13</label>
        <nail x="-748" y="-212"/>
    </transition>
    <transition>
        <source ref="id13"/>
        <target ref="id6"/>
        <label kind="assignment" x="-739" y="-187">fal(1)</label>
        <label kind="probability" x="-765" y="-187">9</label>
    </transition>
    <transition>
        <source ref="id7"/>
        <target ref="id13"/>
        <label kind="guard" x="-867" y="-187">old==0</label>
    </transition>
    <transition>
        <source ref="id14"/>
        <target ref="id7"/>
        <label kind="probability" x="-943" y="-204">87</label>
    </transition>
    <transition>
        <source ref="id8"/>
        <target ref="id14"/>
        <label kind="guard" x="-1028" y="-221">old==1</label>
        <label kind="assignment" x="-1020" y="-195">old=0</label>
    </transition>
    <transition>
        <source ref="id16"/>
        <target ref="id10"/>
        <label kind="assignment" x="-1666" y="-85">juvenile--
</label>

        <label kind="probability" x="-1547" y="-127">78</label>
    </transition>

```

```

        <transition>
            <source ref="id9"/>
            <target ref="id16"/>
            <label kind="guard" x="-1581" y="-170">old==2</label>
        </transition>
    </template>
    <template>
        <name>StartingFal</name>
        <declaration>clock old;</declaration>
        <location id="id17" x="-246" y="-204">
            <name x="-272" y="-238">Death</name>
        </location>
        <location id="id18" x="-612" y="-59">
            <name x="-646" y="-93">ReturnTravel</name>
            <label kind="invariant" x="-646" y="-42">old<math>\leq 0</math></label>
        </location>
        <location id="id19" x="-374" y="-51">
            <name x="-384" y="-85">Reproduce</name>
            <label kind="invariant" x="-365" y="-34">old<math>\leq 0</math></label>
        </location>
        <location id="id20" x="-731" y="-59">
            <name x="-765" y="-110">Mature_rep</name>
            <label kind="invariant" x="-765" y="-33">old<math>\leq 0</math></label>
        </location>
        <location id="id21" x="-246" y="-85">
            <name x="-340" y="-127">ToMadagascar</name>
            <label kind="invariant" x="-256" y="-68">old<math>\leq 1</math></label>
        </location>
        <branchpoint id="id22" x="-246" y="-144">
        </branchpoint>
        <branchpoint id="id23" x="-518" y="-51">
        </branchpoint>
        <init ref="id20"/>
        <transition>
            <source ref="id22"/>
            <target ref="id17"/>
            <label kind="assignment" x="-323" y="-204">falcon--
</label>

            <label kind="probability" x="-238" y="-170">13</label>
        </transition>
        <transition>
            <source ref="id22"/>
            <target ref="id18"/>
            <label kind="assignment" x="-484" y="-187">old=0</label>
            <label kind="probability" x="-314" y="-153">87</label>

```

```

        <nail x="-433" y="-178"/>
    </transition>
    <transition>
        <source ref="id21"/>
        <target ref="id22"/>
        <label kind="guard" x="-238" y="-119">old==1</label>
    </transition>
    <transition>
        <source ref="id18"/>
        <target ref="id23"/>
        <label kind="guard" x="-586" y="-51">old==0</label>
    </transition>
    <transition>
        <source ref="id20"/>
        <target ref="id18"/>
        <label kind="guard" x="-697" y="-85">old==0</label>
        <label kind="assignment" x="-705" y="-59">falcon++</label>
    </transition>
    <transition>
        <source ref="id19"/>
        <target ref="id21"/>
        <label kind="guard" x="-365" y="-102">old==0</label>
        <label kind="synchronisation" x="-323" y="-
59">falconC!</label>
    </transition>
    <transition>
        <source ref="id23"/>
        <target ref="id19"/>
        <label kind="assignment" x="-459" y="-43">fal(2)</label>
        <label kind="probability" x="-484" y="-43">25</label>
        <nail x="-425" y="17"/>
    </transition>
    <transition>
        <source ref="id23"/>
        <target ref="id19"/>
        <label kind="assignment" x="-467" y="16">fal(3)</label>
        <label kind="probability" x="-467" y="0">53</label>
        <nail x="-433" y="76"/>
    </transition>
    <transition>
        <source ref="id23"/>
        <target ref="id19"/>
        <label kind="probability" x="-510" y="-102">13</label>
        <nail x="-459" y="-102"/>
    </transition>

```

```

        <transition>
            <source ref="id23"/>
            <target ref="id19"/>
            <label kind="assignment" x="-450" y="-77">fal(1)</label>
            <label kind="probability" x="-476" y="-77">9</label>
        </transition>
    </template>
    <system>M1=StartingFal();
M2=StartingFal();M3=StartingFal();M4=StartingFal();M5=StartingFal();
M6=StartingFal();M7=StartingFal();M8=StartingFal();M9=StartingFal();
M10=StartingFal();M11=StartingFal();M12=StartingFal();M13=StartingFal();
M14=StartingFal();M15=StartingFal();M16=StartingFal();M17=StartingFal();
M18=StartingFal();
M19=StartingFal();
M20=StartingFal();
M21=StartingFal();
F1=Falcon(0);F2=Falcon(1);F3=Falcon(2);F4=Falcon(3);F5=Falcon(4);F6=Falcon(5);
F7=Falcon(6);F8=Falcon(7);F9=Falcon(8);F10=Falcon(9);F11=Falcon(10);F12=Falcon
(11);F13=Falcon(12);F14=Falcon(13);F15=Falcon(14);F16=Falcon(15);F17=Falcon(16
);F18=Falcon(17);F19=Falcon(18);F20=Falcon(19);F21=Falcon(20);F22=Falcon(21);F
23=Falcon(22);F24=Falcon(23);F25=Falcon(24);F26=Falcon(25);F27=Falcon(26);F28=
Falcon(27);F29=Falcon(28);F30=Falcon(29);F31=Falcon(30);F32=Falcon(31);F33=Fal
con(32);F34=Falcon(33);F35=Falcon(34);F36=Falcon(35);F37=Falcon(36);F38=Falcon
(37);F39=Falcon(38);F40=Falcon(39);F41=Falcon(40);F42=Falcon(41);F43=Falcon(42
);F44=Falcon(43);F45=Falcon(44);F46=Falcon(45);F47=Falcon(46);F48=Falcon(47);F4
9=Falcon(48);F50=Falcon(49);F51=Falcon(50);F52=Falcon(51);F53=Falcon(52);F54=F
alcon(53);F55=Falcon(54);F56=Falcon(55);F57=Falcon(56);
F58=Falcon(57);F59=Falcon(58);F60=Falcon(59);F61=Falcon(60);F62=Falcon(61);F63
=Falcon(62);F64=Falcon(63);F65=Falcon(64);F66=Falcon(65);F67=Falcon(66);F68=Fa
lcon(67);F69=Falcon(68);F70=Falcon(69);F71=Falcon(70);F72=Falcon(71);F73=Falco
n(72);F74=Falcon(73);F75=Falcon(74);F76=Falcon(75);F77=Falcon(76);F78=Falcon(7
7);F79=Falcon(78);
F80=Falcon(79);F81=Falcon(80);F82=Falcon(81);F83=Falcon(82);F84=Falcon(83);F85
=Falcon(84);F86=Falcon(85);F87=Falcon(86);F88=Falcon(87);F89=Falcon(88);F90=Fa
lcon(89);F91=Falcon(90);F92=Falcon(91);F93=Falcon(92);F94=Falcon(93);F95=Falco
n(94);F96=Falcon(95);F97=Falcon(96);F98=Falcon(97);F99=Falcon(98);F100=Falcon(
99);F101=Falcon(100);F102=Falcon(101);F103=Falcon(102);F104=Falcon(103);F105=F
alcon(104);F106=Falcon(105);F107=Falcon(106);F108=Falcon(107);F109=Falcon(108)
;F110=Falcon(109);F111=Falcon(110);F112=Falcon(111);F113=Falcon(112);F114=Falc
on(113);F115=Falcon(114);F116=Falcon(115);F117=Falcon(116);F118=Falcon(117);F1
19=Falcon(118);F120=Falcon(119);F121=Falcon(120);F122=Falcon(121);F123=Falcon(
122);F124=Falcon(123);F125=Falcon(124);F126=Falcon(125);F127=Falcon(126);F128=
Falcon(127);F129=Falcon(128);F130=Falcon(129);F131=Falcon(130);F132=Falcon(131
);F133=Falcon(132);F134=Falcon(133);F135=Falcon(134);F136=Falcon(135);F137=Fal
con(136);F138=Falcon(137);

```

F139=Falcon(138);F140=Falcon(139);F141=Falcon(140);F142=Falcon(141);F143=Falcon(142);F144=Falcon(143);F145=Falcon(144);F146=Falcon(145);F147=Falcon(146);F148=Falcon(147);F149=Falcon(148);F150=Falcon(149);F151=Falcon(150);F152=Falcon(151);F153=Falcon(152);F154=Falcon(153);F155=Falcon(154);F156=Falcon(155);F157=Falcon(156);F158=Falcon(157);F159=Falcon(158);F160=Falcon(159);F161=Falcon(160);F162=Falcon(161);F163=Falcon(162);F164=Falcon(163);F165=Falcon(164);F166=Falcon(165);F167=Falcon(166);F168=Falcon(167);F169=Falcon(168);F170=Falcon(169);F171=Falcon(170);F172=Falcon(171);F173=Falcon(172);F174=Falcon(173);F175=Falcon(174);F176=Falcon(175);F177=Falcon(176);F178=Falcon(177);F179=Falcon(178);F180=Falcon(179);F181=Falcon(180);F182=Falcon(181);F183=Falcon(182);F184=Falcon(183);F185=Falcon(184);F186=Falcon(185);F187=Falcon(186);F188=Falcon(187);F189=Falcon(188);F190=Falcon(189);F191=Falcon(190);F192=Falcon(191);F193=Falcon(192);F194=Falcon(193);F195=Falcon(194);F196=Falcon(195);F197=Falcon(196);F198=Falcon(197);F199=Falcon(198);F200=Falcon(199);F201=Falcon(200);F202=Falcon(201);F203=Falcon(202);F204=Falcon(203);F205=Falcon(204);F206=Falcon(205);F207=Falcon(206);F208=Falcon(207);F209=Falcon(208);F210=Falcon(209);F211=Falcon(210);F212=Falcon(211);F213=Falcon(212);F214=Falcon(213);F215=Falcon(214);F216=Falcon(215);F217=Falcon(216);F218=Falcon(217);F219=Falcon(218);F220=Falcon(219);F221=Falcon(220);F222=Falcon(221);F223=Falcon(222);F224=Falcon(223);F225=Falcon(224);F226=Falcon(225);F227=Falcon(226);F228=Falcon(227);F229=Falcon(228);F230=Falcon(229);F231=Falcon(230);F232=Falcon(231);F233=Falcon(232);F234=Falcon(233);F235=Falcon(234);F236=Falcon(235);F237=Falcon(236);F238=Falcon(237);F239=Falcon(238);F240=Falcon(239);F241=Falcon(240);F242=Falcon(241);F243=Falcon(242);F244=Falcon(243);F245=Falcon(244);F246=Falcon(245);F247=Falcon(246);F248=Falcon(247);F249=Falcon(248);F250=Falcon(249);F251=Falcon(250);F252=Falcon(251);F253=Falcon(252);F254=Falcon(253);F255=Falcon(254);F256=Falcon(255);F257=Falcon(256);F258=Falcon(257);F259=Falcon(258);F260=Falcon(259);F261=Falcon(260);F262=Falcon(261);F263=Falcon(262);F264=Falcon(263);F265=Falcon(264);F266=Falcon(265);F267=Falcon(266);F268=Falcon(267);F269=Falcon(268);F270=Falcon(269);F271=Falcon(270);F272=Falcon(271);F273=Falcon(272);F274=Falcon(273);F275=Falcon(274);F276=Falcon(275);F277=Falcon(276);F278=Falcon(277);F279=Falcon(278);F280=Falcon(279);F281=Falcon(280);F282=Falcon(281);F283=Falcon(282);F284=Falcon(283);F285=Falcon(284);F286=Falcon(285);F287=Falcon(286);F288=Falcon(287);F289=Falcon(288);F290=Falcon(289);F291=Falcon(290);F292=Falcon(291);F293=Falcon(292);F294=Falcon(293);F295=Falcon(294);F296=Falcon(295);F297=Falcon(296);F298=Falcon(297);F299=Falcon(298);F300=Falcon(299);F301=Falcon(300);F302=Falcon(301);F303=Falcon(302);F304=Falcon(303);F305=Falcon(304);F306=Falcon(305);F307=Falcon(306);F308=Falcon(307);F309=Falcon(308);F310=Falcon(309);F311=Falcon(310);F312=Falcon(311);F313=Falcon(312);F314=Falcon(313);F315=Falcon(314);F316=Falcon(315);F317=Falcon(316);F318=Falcon(317);F319=Falcon(318);F320=Falcon(319);F321=Falcon(320);F322=Falcon(321);F323=Falcon(322);F324=Falcon(323);F325=Falcon(324);F326=Falcon(325);F327=Falcon(326);F328=Falcon(327);F329=Falcon(328);F330=Falcon(329);F331=Falcon(330);F332=Falcon(331);F333=Falcon(332);F334=Falcon(333);F335=Falcon(334);F336=Falcon(335);F337=Falcon(336);F338=Falcon(337);F339=Falcon(338);F340=Falcon(339);F341=Falcon(340);F342=Falcon(341);F343=Falcon(342);F344=Falcon(343);F

F345=Falcon (344) ; F346=Falcon (345) ; F347=Falcon (346) ; F348=Falcon (347) ; F349=Falcon (348) ; F350=Falcon (349) ; F351=Falcon (350) ; F352=Falcon (351) ; F353=Falcon (352) ; F354=Falcon (353) ; F355=Falcon (354) ; F356=Falcon (355) ; F357=Falcon (356) ; F358=Falcon (357) ; F359=Falcon (358) ; F360=Falcon (359) ; F361=Falcon (360) ; F362=Falcon (361) ; F363=Falcon (362) ; F364=Falcon (363) ; F365=Falcon (364) ; F366=Falcon (365) ; F367=Falcon (366) ; F368=Falcon (367) ; F369=Falcon (368) ; F370=Falcon (369) ; F371=Falcon (370) ; F372=Falcon (371) ; F373=Falcon (372) ; F374=Falcon (373) ; F375=Falcon (374) ; F376=Falcon (375) ; F377=Falcon (376) ; F378=Falcon (377) ; F379=Falcon (378) ; F380=Falcon (379) ; F381=Falcon (380) ; F382=Falcon (381) ; F383=Falcon (382) ; F384=Falcon (383) ; F385=Falcon (384) ; F386=Falcon (385) ; F387=Falcon (386) ; F388=Falcon (387) ; F389=Falcon (388) ; F390=Falcon (389) ; F391=Falcon (390) ; F392=Falcon (391) ; F393=Falcon (392) ; F394=Falcon (393) ; F395=Falcon (394) ; F396=Falcon (395) ; F397=Falcon (396) ; F398=Falcon (397) ; F399=Falcon (398) ; F400=Falcon (399) ; F401=Falcon (400) ; F402=Falcon (401) ; F403=Falcon (402) ; F404=Falcon (403) ; F405=Falcon (404) ; F406=Falcon (405) ; F407=Falcon (406) ; F408=Falcon (407) ; F409=Falcon (408) ; F410=Falcon (409) ; F411=Falcon (410) ; F412=Falcon (411) ; F413=Falcon (412) ; F414=Falcon (413) ; F415=Falcon (414) ; F416=Falcon (415) ; F417=Falcon (416) ; F418=Falcon (417) ; F419=Falcon (418) ; F420=Falcon (419) ; F421=Falcon (420) ; F422=Falcon (421) ; F423=Falcon (422) ; F424=Falcon (423) ; F425=Falcon (424) ; F426=Falcon (425) ; F427=Falcon (426) ; F428=Falcon (427) ; F429=Falcon (428) ; F430=Falcon (429) ; F431=Falcon (430) ; F432=Falcon (431) ; F433=Falcon (432) ; F434=Falcon (433) ; F435=Falcon (434) ; F436=Falcon (435) ; F437=Falcon (436) ; F438=Falcon (437) ; F439=Falcon (438) ; F440=Falcon (439) ; F441=Falcon (440) ; F442=Falcon (441) ; F443=Falcon (442) ; F444=Falcon (443) ; F445=Falcon (444) ; F446=Falcon (445) ; F447=Falcon (446) ; F448=Falcon (447) ; F449=Falcon (448) ; F450=Falcon (449) ; F451=Falcon (450) ; F452=Falcon (451) ; F453=Falcon (452) ; F454=Falcon (453) ; F455=Falcon (454) ; F456=Falcon (455) ; F457=Falcon (456) ; F458=Falcon (457) ; F459=Falcon (458) ; F460=Falcon (459) ; F461=Falcon (460) ; F462=Falcon (461) ; F463=Falcon (462) ; F464=Falcon (463) ; F465=Falcon (464) ; F466=Falcon (465) ; F467=Falcon (466) ; F468=Falcon (467) ; F469=Falcon (468) ; F470=Falcon (469) ; F471=Falcon (470) ; F472=Falcon (471) ; F473=Falcon (472) ; F474=Falcon (473) ; F475=Falcon (474) ; F476=Falcon (475) ; F477=Falcon (476) ; F478=Falcon (477) ; F479=Falcon (478) ; F480=Falcon (479) ; F481=Falcon (480) ; F482=Falcon (481) ; F483=Falcon (482) ; F484=Falcon (483) ; F485=Falcon (484) ; F486=Falcon (485) ; F487=Falcon (486) ; F488=Falcon (487) ; F489=Falcon (488) ; F490=Falcon (489) ; F491=Falcon (490) ; F492=Falcon (491) ; F493=Falcon (492) ; F494=Falcon (493) ; F495=Falcon (494) ; F496=Falcon (495) ; F497=Falcon (496) ; F498=Falcon (497) ; F499=Falcon (498) ; F500=Falcon (499) ;

system

M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,M11,M12,M13,M14,M15,M16,M17,M18,M19,M20,M21,F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12,F13,F14,F15,F16,F17,F18,F19,F20,F21,F22,F23,F24,F25,F26,F27,F28,F29,F30,F31,F32,F33,F34,F35,F36,F37,F38,F39,F40,F41,F42,F43,F44,F45,F46,F47,F48,F49,F50,F51,F52,F53,F54,F55,F56,F57,F58,F59,F60,F61,F62,F63,F64,F65,F66,F67,F68,F69,F70,F71,F72,F73,F74,F75,F76,F77,F78,F79,F80,F81,F82,F83,F84,F85,F86,F87,F88,F89,F90,F91,F92,F93,F94,F95,F96,F97,F98,F99,F100,F101,F102,F103,F104,F105,F106,F107,F108,F109,F110,F111,F112,F113,F114,F115,F116,F117,F118,F119,F120,F121,F122,F123,F124,F125,F126,F127,F128,F129,F130,F131,F132,F133,F134,F135,F136,F137,F138,F139,F140,F141,F142,F143,F144,F145,F146,F147,

F148,F149,F150,F151,F152,F153,F154,F155,F156,F157,F158,F159,F160,F161,F162,F163,F164,F165,F166,F167,F168,F169,F170,F171,F172,F173,F174,F175,F176,F177,F178,F179,F180,F181,F182,F183,F184,F185,F186,F187,F188,F189,F190,F191,F192,F193,F194,F195,F196,F197,F198,F199,F200,F201,F202,F203,F204,F205,F206,F207,F208,F209,F210,F211,F212,F213,F214,F215,F216,F217,F218,F219,F220,F221,F222,F223,F224,F225,F226,F227,F228,F229,F230,F231,F232,F233,F234,F235,F236,F237,F238,F239,F240,F241,F242,F243,F244,F245,F246,F247,F248,F249,F250,F251,F252,F253,F254,F255,F256,F257,F258,F259,F260,F261,F262,F263,F264,F265,F266,F267,F268,F269,F270,F271,F272,F273,F274,F275,F276,F277,F278,F279,F280,F281,F282,F283,F284,F285,F286,F287,F288,F289,F290,F291,F292,F293,F294,F295,F296,F297,F298,F299,F300,F301,F302,F303,F304,F305,F306,F307,F308,F309,F310,F311,F312,F313,F314,F315,F316,F317,F318,F319,F320,F321,F322,F323,F324,F325,F326,F327,F328,F329,F330,F331,F332,F333,F334,F335,F336,F337,F338,F339,F340,F341,F342,F343,F344,F345,F346,F347,F348,F349,F350,F351,F352,F353,F354,F355,F356,F357,F358,F359,F360,F361,F362,F363,F364,F365,F366,F367,F368,F369,F370,F371,F372,F373,F374,F375,F376,F377,F378,F379,F380,F381,F382,F383,F384,F385,F386,F387,F388,F389,F390,F391,F392,F393,F394,F395,F396,F397,F398,F399,F400,F401,F402,F403,F404,F405,F406,F407,F408,F409,F410,F411,F412,F413,F414,F415,F416,F417,F418,F419,F420,F421,F422,F423,F424,F425,F426,F427,F428,F429,F430,F431,F432,F433,F434,F435,F436,F437,F438,F439,F440,F441,F442,F443,F444,F445,F446,F447,F448,F449,F450,F451,F452,F453,F454,F455,F456,F457,F458,F459,F460,F461,F462,F463,F464,F465,F466,F467,F468,F469,F470,F471,F472,F473,F474,F475,F476,F477,F478,F479,F480,F481,F482,F483,F484,F485,F486,F487,F488,F489,F490,F491,F492,F493,F494,F495,F496,F497,F498,F499,F500;</system>

<queries>

<query>

<formula>E[<=1;100] (max:flying)

</formula>

<comment>

</comment>

</query>

<query>

<formula>E[<=2;100] (max:falcon)

</formula>

<comment>

</comment>

</query>

<query>

<formula>E[<=2;100] (max:flying)

</formula>

<comment>

</comment>

</query>

<query>

<formula>E[<=3;100] (max:falcon)

</formula>

```

        <comment>
        </comment>
</query>
<query>
    <formula>E[&lt;=3;100] (max:flying)
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>E[&lt;=4;100] (max:falcon)
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>E[&lt;=4;100] (max:flying)
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>E[&lt;=5;100] (max:falcon)
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>E[&lt;=5;100] (max:flying)
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>E[&lt;=6;100] (max:falcon)
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>E[&lt;=6;100] (max:flying)
    </formula>
    <comment>
    </comment>
</query>

```

```

<query>
    <formula>E[&lt;=7;100] (max:falcon)
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>E[&lt;=7;100] (max:flying)
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>E[&lt;=8;100] (max:falcon)
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>E[&lt;=8;100] (max:flying)
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>E[&lt;=9;100] (max:falcon)
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>E[&lt;=9;100] (max:flying)
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>E[&lt;=10;100] (max:falcon)
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>E[&lt;=10;100] (max:flying)
    </formula>

```

```

        <comment>
        </comment>
    </query>
    <query>
        <formula>E[&lt;=11;100] (max:falcon)
        </formula>
        <comment>
        </comment>
    </query>
    <query>
        <formula>E[&lt;=11;100] (max:flying)
        </formula>
        <comment>
        </comment>
    </query>
    <query>
        <formula>E[&lt;=12;100] (max:falcon)
        </formula>
        <comment>
        </comment>
    </query>
    <query>
        <formula>E[&lt;=12;100] (max:flying)
        </formula>
        <comment>
        </comment>
    </query>
    <query>
        <formula>E[&lt;=13;100] (max:falcon)
        </formula>
        <comment>
        </comment>
    </query>
    <query>
        <formula>E[&lt;=13;100] (max:flying)
        </formula>
        <comment>
        </comment>
    </query>
</queries>
</nta>

```

Παράρτημα Ε

Κώδικας σε xml σε UPPAAL SMC για την μοντελοποίηση της προσέγγισης Δυναμικής Δημιουργία Στοιχείων

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE nta PUBLIC "-//Uppaal Team//DTD Flat System 1.1//EN"
'http://www.it.uu.se/research/group/darts/uppaal/flat-1_2.dtd'>
<nta>
    <declaration>// Place global declarations here.
    int falcon,egg,nestling,flying,juvenile=0;

    int num=0;
    int start_falcon=21;
    int start_nestling=0;
    int start_egg=0;

    dynamic Falcon(int num);
    dynamic Egg(int num);
    dynamic Nestling(int num);

    int rats,sterility,human,sun,preditors,canibalism,deseases;</declaration>
    <template>
        <name>Egg</name>
        <parameter>int num</parameter>
        <declaration>clock old;
    int id_egg;</declaration>
        <location id="id0" x="-645" y="-69">
            <name x="-655" y="-103">born</name>
            <label kind="invariant" x="-655" y="-52">old<=0</label>
        </location>
        <location id="id1" x="-289" y="-76">
            <name x="-299" y="-110">Death</name>
        </location>
        <location id="id2" x="-416" y="-76">
            <name x="-441" y="-119">Hatch</name>
            <label kind="invariant" x="-426" y="-59">old<=0</label>
        </location>
        <location id="id3" x="-731" y="-68">
            <name x="-767" y="-102">Start</name>
            <committed/>
        </location>
        <branchpoint id="id4" x="-551" y="-69">
    </branchpoint>
```

```

<init ref="id3"/>
<transition>
    <source ref="id4"/>
    <target ref="id2"/>
    <label kind="probability" x="-527" y="-59">14</label>
    <nail x="-476" y="-51"/>
</transition>
<transition>
    <source ref="id4"/>
    <target ref="id2"/>
    <label kind="assignment" x="-595" y="-144">spawn
Nestling(num++),nestling++</label>
    <label kind="probability" x="-535" y="-110">86</label>
    <nail x="-476" y="-119"/>
</transition>
<transition>
    <source ref="id0"/>
    <target ref="id4"/>
    <label kind="guard" x="-619" y="-94">old==0</label>
</transition>
<transition>
    <source ref="id3"/>
    <target ref="id0"/>
    <label kind="assignment" x="-755" y="-
119">old=0,id_egg=num</label>
</transition>
<transition>
    <source ref="id2"/>
    <target ref="id1"/>
    <label kind="guard" x="-365" y="-110">old==0</label>
    <label kind="assignment" x="-357" y="-68">egg--,
exit()</label>
</transition>
</template>
<template>
    <name>Nestling</name>
    <parameter>int num</parameter>
    <declaration>clock old;
int id_nes;</declaration>
    <location id="id5" x="-221" y="34">
        <name x="-246" y="-9">Fly</name>
        <label kind="invariant" x="-231" y="51">old<=0</label>
    </location>
    <location id="id6" x="-450" y="42">
        <name x="-460" y="8">Born</name>

```

```

        <label kind="invariant" x="-460" y="59">old<math>\leq 0</math></label>
    </location>
    <location id="id7" x="-119" y="34">
        <name x="-153" y="-17">Flying_F</name>
        <label kind="invariant" x="-161"
y="51">old<math>\leq 0.001</math></label>
    </location>
    <location id="id8" x="-536" y="43">
        <name x="-572" y="9">Start</name>
        <committed/>
    </location>
    <location id="id9" x="-25" y="34">
        <name x="-34" y="-9">Death</name>
    </location>
    <branchpoint id="id10" x="-356" y="42">
</branchpoint>
    <init ref="id8"/>
    <transition>
        <source ref="id7"/>
        <target ref="id9"/>
        <label kind="guard" x="-102" y="0">old==0.001</label>
        <label kind="assignment" x="-85" y="42">nestling--,
exit()</label>
    </transition>
    <transition>
        <source ref="id10"/>
        <target ref="id5"/>
        <label kind="probability" x="-323" y="25">26</label>
        <nail x="-289" y="59"/>
    </transition>
    <transition>
        <source ref="id10"/>
        <target ref="id5"/>
        <label kind="assignment" x="-374" y="-42">spawn
Falcon(num++),flying++</label>
        <label kind="probability" x="-349" y="1">74</label>
        <nail x="-280" y="-8"/>
    </transition>
    <transition>
        <source ref="id6"/>
        <target ref="id10"/>
        <label kind="guard" x="-424" y="17">old==0</label>
    </transition>
    <transition>
        <source ref="id8"/>

```

```

        <target ref="id6"/>
        <label kind="assignment" x="-560" y="-
8">old=0,id_nes=num</label>
    </transition>
    <transition>
        <source ref="id5"/>
        <target ref="id7"/>
        <label kind="guard" x="-195" y="8">old==0</label>
    </transition>
</template>
<template>
    <name>Falcon</name>
    <parameter>int num</parameter>
    <declaration>clock old;
int id_f;
</declaration>
    <location id="id11" x="-331" y="-111">
        <name x="-374" y="-153">Reproduce1</name>
        <label kind="invariant" x="-340" y="-
102">old<=0</label>
    </location>
    <location id="id12" x="-382" y="68">
        <name x="-425" y="59">End</name>
    </location>
    <location id="id13" x="299" y="-68">
        <name x="281" y="-102">Mature_rep</name>
        <label kind="invariant" x="281" y="-43">old<=0</label>
    </location>
    <location id="id14" x="-382" y="8">
        <name x="-433" y="17">Death</name>
        <label kind="invariant" x="-374" y="17">old<=0</label>
    </location>
    <location id="id15" x="136" y="-144">
        <name x="102" y="-204">Reproduce2</name>
        <label kind="invariant" x="128" y="-119">old<=1</label>
    </location>
    <location id="id16" x="-501" y="-136">
        <name x="-552" y="-144">Adult</name>
        <label kind="invariant" x="-552" y="-
119">old<=3</label>
    </location>
    <location id="id17" x="-501" y="-238">
        <name x="-511" y="-272">Start</name>
        <committed/>
    </location>

```



```

<location id="id18" x="680" y="-51">
    <name x="680" y="-85">R2_finished</name>
    <label kind="invariant" x="697" y="-60">old<math>\leq 0</math></label>
</location>
<branchpoint id="id19" x="-408" y="-68">
</branchpoint>
<branchpoint id="id20" x="247" y="-102">
</branchpoint>
<branchpoint id="id21" x="-271" y="-102">
</branchpoint>
<branchpoint id="id22" x="366" y="-26">
</branchpoint>
<init ref="id17"/>
<transition>
    <source ref="id17"/>
    <target ref="id16"/>
    <label kind="assignment" x="-488" y="-208">old=0,id_f=num,juvenile++</label>
</transition>
<transition>
    <source ref="id16"/>
    <target ref="id19"/>
    <label kind="guard" x="-467" y="-161">old==2</label>
    <label kind="assignment" x="-433" y="-136">juvenile--,old=0</label>
    <nail x="-450" y="-136"/>
</transition>
<transition>
    <source ref="id19"/>
    <target ref="id14"/>
    <label kind="probability" x="-391" y="-42">856</label>
</transition>
<transition>
    <source ref="id16"/>
    <target ref="id19"/>
    <label kind="guard" x="-484" y="-93">old==3</label>
    <label kind="assignment" x="-510" y="-34">juvenile--,old=0</label>
</transition>
<transition>
    <source ref="id11"/>
    <target ref="id21"/>
    <label kind="guard" x="-306" y="-136">old==0</label>
</transition>
<transition>

```

```

        <source ref="id19"/>
        <target ref="id11"/>
        <label kind="assignment" x="-348" y="-85">falcon++,
old=0</label>
        <label kind="probability" x="-391" y="-110">154</label>
    </transition>
    <transition>
        <source ref="id18"/>
        <target ref="id15"/>
        <label kind="guard" x="595" y="-213">old==0</label>
        <label kind="assignment" x="587" y="-230">old=0</label>
        <nail x="518" y="-272"/>
    </transition>
    <transition>
        <source ref="id21"/>
        <target ref="id15"/>
        <label kind="assignment" x="-195" y="-43">spawn
Egg(num++),egg++,
spawn Egg(num++),egg++,
spawn Egg(num++),egg++</label>
        <label kind="probability" x="-271" y="-77">35</label>
        <nail x="-263" y="-85"/>
        <nail x="-186" y="-43"/>
        <nail x="-33" y="-43"/>
    </transition>
    <transition>
        <source ref="id21"/>
        <target ref="id15"/>
        <label kind="probability" x="-280" y="-162">40</label>
        <nail x="-212" y="-222"/>
        <nail x="-76" y="-230"/>
    </transition>
    <transition>
        <source ref="id22"/>
        <target ref="id18"/>
        <label kind="probability" x="400" y="-119">30</label>
        <nail x="510" y="-213"/>
    </transition>
    <transition>
        <source ref="id22"/>
        <target ref="id18"/>
        <label kind="assignment" x="459" y="-17">spawn
Egg(num++),egg++,
spawn Egg(num++),egg++,
spawn Egg(num++),egg++</label>

```

```

        <label kind="probability" x="425" y="-17">42</label>
        <nail x="544" y="76"/>
    </transition>
    <transition>
        <source ref="id22"/>
        <target ref="id18"/>
        <label kind="assignment" x="451" y="-119">spawn
Egg(num++),egg++</label>
        <label kind="probability" x="408" y="-85">10</label>
        <nail x="526" y="-102"/>
    </transition>
    <transition>
        <source ref="id21"/>
        <target ref="id15"/>
        <label kind="assignment" x="-203" y="-153">spawn
Egg(num++),egg++</label>
        <label kind="probability" x="-237" y="-145">9</label>
        <nail x="-152" y="-170"/>
    </transition>
    <transition>
        <source ref="id14"/>
        <target ref="id12"/>
        <label kind="guard" x="-433" y="34">old==0</label>
        <label kind="assignment" x="-365" y="34">exit()</label>
    </transition>
    <transition>
        <source ref="id13"/>
        <target ref="id22"/>
        <label kind="guard" x="317" y="-81">old==0</label>
    </transition>
    <transition>
        <source ref="id20"/>
        <target ref="id13"/>
        <label kind="probability" x="264" y="-119">87</label>
    </transition>
    <transition>
        <source ref="id21"/>
        <target ref="id15"/>
        <label kind="assignment" x="-212" y="-102">spawn
Egg(num++),egg++,
spawn Egg(num++),egg++</label>
        <label kind="probability" x="-246" y="-102">16</label>
    </transition>
    <transition>
        <source ref="id20"/>

```

```

        <target ref="id14"/>
        <label kind="assignment" x="-8" y="8">falcon--</label>
        <label kind="probability" x="221" y="-68">13</label>
        <nail x="-17" y="110"/>
    </transition>
    <transition>
        <source ref="id15"/>
        <target ref="id20"/>
        <label kind="guard" x="204" y="-145">old==1</label>
        <label kind="assignment" x="221" y="-136">old=0</label>
    </transition>
    <transition>
        <source ref="id22"/>
        <target ref="id18"/>
        <label kind="assignment" x="459" y="-68">spawn
Egg(num++),egg++,
spawn Egg(num++),egg++</label>
        <label kind="probability" x="408" y="-51">18</label>
        <nail x="493" y="-34"/>
    </transition>
</template>
<template>
    <name>Starting</name>
    <declaration>clock time;</declaration>
    <location id="id23" x="-6307" y="-5967">
        <name x="-6317" y="-6001">Start</name>
        <committed/>
    </location>
    <location id="id24" x="-5797" y="-5967">
        <name x="-5807" y="-6001">end</name>
    </location>
    <location id="id25" x="-5958" y="-5967">
        <name x="-5984" y="-6026">Falcon_finished</name>
        <label kind="invariant" x="-5984" y="-
5950">time<math>\leq 0</math></label>
    </location>
    <location id="id26" x="-6180" y="-5967">
        <name x="-6190" y="-6001">Rep</name>
        <label kind="invariant" x="-6196" y="-
5950">time<math>\leq 0</math></label>
    </location>
    <init ref="id23"/>
    <transition>
        <source ref="id23"/>
        <target ref="id26"/>

```

```

        <label kind="assignment" x="-6281" y="-
5958">time=0</label>
    </transition>
    <transition>
        <source ref="id25"/>
        <target ref="id24"/>
        <label kind="guard" x="-5950" y="-
6001">falcon==start_falcon</label>
    </transition>
    <transition>
        <source ref="id25"/>
        <target ref="id26"/>
        <label kind="guard" x="-6111" y="-
6086">falcon<&lt;start_falcon</label>
        <nail x="-6060" y="-6052"/>
    </transition>
    <transition>
        <source ref="id26"/>
        <target ref="id25"/>
        <label kind="guard" x="-6111" y="-5967">time==0</label>
        <label kind="assignment" x="-6205" y="-5933">spawn
Falcon(num++),falcon++</label>
    </transition>
</template>
<system>// Place template instantiations here.
//Falcon1 = FalconProb();
//Falcon2 = FalconProb();
//Falcon3 = FalconProb();
//Master = StartingProb();
// List one or more processes to be composed into a system.
//system Master,Falcon1,Falcon2,Falcon3;

// Place template instantiations here.
Master = Starting();
// List one or more processes to be composed into a system.
system Master;
</system>
<queries>
    <query>
        <formula>simulate
100[&lt;=100]{reproduce,rats,sterility,human,sun,preditors,canibalism,deseases
,natural}

        </formula>
        <comment>
        </comment>

```

```

</query>
<query>
    <formula>simulate 100[&lt;=100] {[ ] Falcon(0).Rats}
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>E[&lt;=3;1] (max: falcon)
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>Pr [ &lt;=100] ( &lt;&gt;rats&gt;=100)
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>simulate 10[&lt;=100]{falcon}
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>simulate 1000[&lt;=100]{egg}
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>simulate 1[&lt;=100]{falcon}
    </formula>
    <comment>
    </comment>
</query>
<query>
    <formula>simulate 50[&lt;=60]{falcon,egg,clutch}
    </formula>
    <comment>
    </comment>
</query>
</queries>
</nta>

```


Παράρτημα F

Στοιχεία μοντελοποίησης Μαυροπετρίτη UPPAAL SMC για 13 χρόνια

	1	2	3	4	5	6	7	8	9	10	11	12	13
Φωλιές	21	21	23	27	30	32	37	44	51	56	66	71	85
Νεοσσοί που πέταξαν	37	38	40	44	48	50	62	74	81	91	110	123	131
Νεκρά Ενήλικα Πουλιά	5	5	7	10	13	17	21	26	32	38	45	55	63
Νεκρά Νεαρά Πουλιά	0	0	28	54	77	102	135	174	217	270	320	381	445
Σύνολο	58	88	113	118	128	157	181	201	239	254	299	340	386